```
C     **************************
C     * PDE2D 9.2 MAIN PROGRAM *
C     **************************
C     *** 2D PROBLEM SOLVED (COLLOCATION METHOD) ***
C##############################################################################
C     Is double precision mode to be used?  Double precision is recommended   #
C     on 32-bit computers.                                                    #
C                                                                             #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + If double precision mode is used, variables and functions assigned   +#
C     + names beginning with a letter in the range A-H or O-Z will be DOUBLE +#
C     + PRECISION, and you should use double precision constants and FORTRAN +#
C     + expressions throughout; otherwise such variables and functions will  +#
C     + be of type REAL.  In either case, variables and functions assigned   +#
C     + names beginning with I,J,K,L,M or N will be of INTEGER type.         +#
C     +                                                                      +#
C     + It is possible to convert a single precision PDE2D program to double +#
C     + precision after it has been created, using an editor.  Just change   +#
C     + all occurrences of "real" to "double precision"                      +#
C     +                    " tdp" to "dtdp"  (note leading blank)            +#
C     + Any user-written code or routines must be converted "by hand", of    +#
C     + course.  To convert from double to single, reverse the changes.     +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++#
C##############################################################################
      implicit double precision (a-h,o-z)
      parameter (neqnmx=  99)
C##############################################################################
C     NP1GRID = number of P1-grid lines                                       #
C##############################################################################
C-----------------------------------------> INPUT FROM GUI <-------------------
      PARAMETER (NP1GRID = 18)
C##############################################################################
C     NP2GRID = number of P2-grid lines                                       #
C##############################################################################
C-----------------------------------------> INPUT FROM GUI <-------------------
      PARAMETER (NP2GRID = 16)
C##############################################################################
C     How many differential equations (NEQN) are there in your problem?       #
C##############################################################################
      PARAMETER (NEQN = 4)
      parameter (np3grid = 1)
C        DIMENSIONS OF WORK ARRAYS
C        SET TO 1 FOR AUTOMATIC ALLOCATION
      PARAMETER (IRWK8Z=           1)
      PARAMETER (IIWK8Z=           1)
      PARAMETER (NXP8Z=101,NYP8Z=101,KDEG8Z=1,NZP8Z=KDEG8Z+1)
C##############################################################################
C     The solution is normally saved on an NP1+1 by NP2+1 rectangular grid    #
C     of points,                                                              #
C                   P1 = P1A + I*(P1B-P1A)/NP1,    I = 0,...,NP1              #
C                   P2 = P2A + J*(P2B-P2A)/NP2,    J = 0,...,NP2              #
C     Enter values for NP1 and NP2.  Suggested values: NP1=NP2=25.            #
C                                                                             #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + If you want to save the solution at an arbitrary user-specified set  +#
C     + of points, set NP2=0 and NP1+1=number of points.  In this case you   +#
C     + can request tabular output, but no plots can be made.                +#
C     +                                                                      +#
C     + If you set NEAR8Z=1 in the main program, the values saved at each    +#
C     + output point will actually be the solution as evaluated at a nearby  +#
C     + collocation point.  For most problems this obviously will produce    +#
C     + less accurate output or plots, but for certain (rare) problems, a    +#
C     + solution component may be much less noisy when plotted only at       +#
C     + collocation points.                                                  +#
```

```
C     +++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++#
C#######################################################################
      PARAMETER (NP1 = 140)
      PARAMETER (NP2 = 63)
C#######################################################################
C     The solution will be saved (for possible postprocessing) at the NSAVE+1 #
C     time points                                                       #
C                         T0 + K*(TF-T0)/NSAVE                          #
C     K=0,...,NSAVE.  Enter a value for NSAVE.                          #
C                                                                       #
C     If a user-specified constant time step is used, NSTEPS must be an       #
C     integer multiple of NSAVE.                                        #
C#######################################################################
      PARAMETER (NSAVE =150)
      common/parm8z/pi,DCA,Umax,EC50 ,Hill,CArest,RC,Aca
     &,OD2,Slope ,Slope2,Width,TN,Flu,Cal,ktn_p,ktn_n,kcal_p
     &,kflu_n,DCal,Dflu ,DU ,OTB ,kcal_n, kflu_p
      dimension p1grid(np1grid),p2grid(np2grid),p3grid(np3grid),p1out8z(
     &0:np1,0:np2),p2out8z(0:np1,0:np2),p3out8z(0:np1,0:np2),p1cross(100
     &),p2cross(100),tout8z(0:nsave),uout8z(0:np1,0:np2,4*neqn,0:nsave),
     &uout(0:np1,0:np2,4,neqn,0:nsave),xres8z(nxp8z),yres8z(nyp8z),zres8
     &z(nzp8z),ures8z(neqn,nxp8z,nyp8z,nzp8z)
      equivalence (uout,uout8z)
      allocatable iwrk8z(:),rwrk8z(:)
C      dimension iwrk8z(iiwk8z),rwrk8z(irwk8z)
      character*40 title
      logical linear,crankn,noupdt,nodist,fillin,evcmpx,adapt,plot,lsqfi
     &t,fdiff,solid,econ8z,ncon8z,restrt,gridid
      common/dtdp14/ sint8z(20),bint8z(20),slim8z(20),blim8z(20)
      common/dtdp15/ evlr8z,ev0r,evli8z,ev0i,evcmpx
      common/dtdp16/ p8z,evr8z(50),evi8z(50)
      common/dtdp19/ toler(neqnmx),adapt
      common/dtdp30/ econ8z,ncon8z
      common/dtdp45/ perdc(neqnmx)
      common/dtdp46/ eps8z,cgtl8z,npmx8z,itype,near8z
      common/dtdp52/ nxa8z,nya8z,nza8z,kd8z
      common/dtdp53/ work8z(nxp8z*nyp8z*nzp8z+9)
      common/dtdp64/ amin8z(4*neqnmx),amax8z(4*neqnmx)
      pi = 4.0*atan(1.d0)
      zr8z = 0.0
      nxa8z = nxp8z
      nya8z = nyp8z
      nza8z = nzp8z
      kd8z = kdeg8z
C#######################################################################
C     If you don't want to read the FINE PRINT, default NPROB.          #
C                                                                       #
C     ++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) +++++++++++++#
C     + If you want to solve several similar problems in the same run, set   +#
C     + NPROB equal to the number of problems you want to solve.  Then NPROB +#
C     + loops through the main program will be done, with IPROB=1,...,NPROB, +#
C     + and you can make the problem parameters vary with IPROB.  NPROB      +#
C     + defaults to 1.                                                  +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++#
C#######################################################################

       NPROB = 1
       do 78755 iprob=1,nprob


C#######################################################################
C     You may now define global parameters, which may be referenced in any    #
C     of the "FORTRAN expressions" you input throughout the rest of this      #
C     interactive session.  You will be prompted alternately for parameter    #
```

```
C     names and their values; enter a blank name when you are finished.      #
C                                                                             #
C     Parameter names are valid FORTRAN variable names, starting in           #
C     column 1.  Thus each name consists of 1 to 6 alphanumeric characters,   #
C     the first of which must be a letter.  If the first letter is in the     #
C     range I-N, the parameter must be an integer.                            #
C                                                                             #
C     Parameter values are either FORTRAN constants or FORTRAN expressions    #
C     involving only constants and global parameters defined on earlier       #
C     lines.  They may also be functions of the problem number IPROB, if      #
C     you are solving several similar problems in one run (NPROB > 1).  Note  #
C     that you are defining global CONSTANTS, not functions; e.g., parameter  #
C     values may not reference any of the independent or dependent variables  #
C     of your problem.                                                        #
C                                                                             #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + If you define other parameters here later, using an editor, you must +#
C     + add them to COMMON block /PARM8Z/ everywhere this block appears, if  +#
C     + they are to be "global" parameters.                                  +#
C     +                                                                      +#
C     + The variable PI is already included as a global parameter, with an   +#
C     + accurate value 3.14159...                                            +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++++#
C##############################################################################
C------------------------------------------> INPUT FROM GUI <------------------

C----- Diffusion coefficient of Calcium ion ----------------------------------
      Umax   =
     & 200
C----- Median effective Ca2+ concentration -----------------------------------
      EC50   =
     & 0.2
C----- Hill coefficient ------------------------------------------------------
      Hill   =
     & 2
C----- diffusion coefficients for Ca2+ --- -----------------------------------
      DCA    =
     & 600
C----- Resting Calcium Concentration -----------------------------------------
      CArest =
     & 0.1
C----- rate constant of activation -------------------------------------------
      RC     =
     & 2
C-----maximum amplitude of Ca2+ flux------------------------------------------
      Aca    =
     & 180
C----- variable time offset  -------------------------------------------------
      DU     =
     & 0
C----- Time base -------------------------------------------------------------
      OTB    =
     & 100
C----- Slope1-----------------------------------------------------------------
      Slope  =
     & 44
C----- Slope2 ----------------------------------------------------------------
      Slope2 =
     & 0.06*Slope
C----- release pulse modulator -----------------------------------------------
      Pulse  =
     & 40
C----- Deactivating time constant --------------------------------------------
      OD2    =
     & OTB*Slope/Pulse
C----- Troponin C Concentration ------------------------------------------------
      TN     =
     & 40
C----- Fluo-4 concentration ----------------------------------------------------
      Flu    =
     & 100
C----- Calmodulin Concentration ------------------------------------------------
      Cal    =
     & 24
C----- On rate of Ca2+-Troponin C-----------------------------------------------
      ktn_p  =
     & 39
C----- Off rate of Ca2+-Troponin C ---------------------------------------------
      ktn_n  =
     & 20
C-----On rate Ca2+-Calmodulin --------------------------------------------------
      kcal_p =
     & 125
C----- Off rate Ca2+-Calmodulin ------------------------------------------------
      kcal_n =
     & 297.5
C----- On rate Ca2+-Fluo-4 -----------------------------------------------------
      kflu_p =
     & 230
C-----Off rate Ca2+-Fluo-4 -----------------------------------------------------
      kflu_n =
     & 170
C----- diffusion coefficients for Ca-CaM --------------------------------------
      DCal   =
     & 25
C----- diffusion coefficients for Ca-Flou4 ------------------------------------
      Dflu   =
     & 100



C###############################################################################
C     A collocation finite element method is used, with bi-cubic Hermite      #
C     basis functions on the elements (small rectangles) defined by the grid  #
C     points:                                                                 #
C               P1GRID(1),...,P1GRID(NP1GRID)                                 #
C               P2GRID(1),...,P2GRID(NP2GRID)                                 #
C     You will first be prompted for NP1GRID, the number of P1-grid points,   #
C     then for P1GRID(1),...,P1GRID(NP1GRID).  Any points defaulted will be   #
C     uniformly spaced between the points you define; the first and last      #
C     points cannot be defaulted.  Then you will be prompted similarly        #
C     for the number and values of the P2-grid points.  The limits on the     #
C     parameters are then:                                                    #
C               P1GRID(1) < P1 < P1GRID(NP1GRID)                              #
C               P2GRID(1) < P2 < P2GRID(NP2GRID)                              #
C                                                                             #
C###############################################################################
      call dtdpwx(p1grid,np1grid,0)
      call dtdpwx(p2grid,np2grid,0)
C        P1GRID DEFINED
C-------------------------------------------> INPUT FROM GUI <-------------------
      P1GRID(1) =
     & 0

C-------------------------------------------> INPUT FROM GUI <-------------------
      P1GRID(NP1GRID) =
     & 30
C        P2GRID DEFINED
```

```
C--------------------------------------------> INPUT FROM GUI <-------------------
      P2GRID(1) =
     & 0

C--------------------------------------------> INPUT FROM GUI <-------------------
      P2GRID(NP2GRID) =
     & 12
C
      p3grid(1) = 0
      call dtdpwx(p1grid,np1grid,1)
      call dtdpwx(p2grid,np2grid,1)
C##############################################################################
C     If you don't want to read the FINE PRINT, enter ISOLVE = 1.             #
C                                                                             #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + The following linear system solvers are available:                   +#
C     +                                                                      +#
C     + 1. Sparse direct method                                              +#
C     +              Harwell Library routine MA27 (used by permission) is    +#
C     +              used to solve the (positive definite) "normal"          +#
C     +              equations A**T*A*x = A**T*b.  The normal equations,     +#
C     +              which are essentially the equations which would result +#
C     +              if a least squares finite element method were used     +#
C     +              instead of a collocation method, are substantially     +#
C     +              more ill-conditioned than the original system Ax = b,  +#
C     +              so it may be important to use high precision if this   +#
C     +              option is chosen.                                       +#
C     + 2. Frontal method                                                    +#
C     +              This is an out-of-core band solver.  If you want to    +#
C     +              override the default number of rows in the buffer (11),+#
C     +              set a new value for NPMX8Z in the main program.        +#
C     + 3. Jacobi conjugate gradient iterative method                        +#
C     +              A preconditioned conjugate gradient iterative method   +#
C     +              is used to solve the (positive definite) normal        +#
C     +              equations.  High precision is also important if this   +#
C     +              option is chosen.  (This solver is MPI-enhanced, if    +#
C     +              MPI is available.)  If you want to override the        +#
C     +              default convergence tolerance, set a new relative      +#
C     +              tolerance CGTL8Z in the main program.                  +#
C     + 4. Local solver (normal equations)                                   +#
C     + 5. Local solver (original equations)                                 +#
C     +              Choose these options ONLY if alterative linear system  +#
C     +              solvers have been installed locally.  See subroutines  +#
C     +              (D)TD3M, (D)TD3N in file (d)subs.f for instructions    +#
C     +              on how to add local solvers.                           +#
C     + 6. MPI-based parallel band solver                                    +#
C     +              This is a parallel solver which runs efficiently on    +#
C     +              multiple processor machines, under MPI.  It is a       +#
C     +              band solver, with the matrix distributed over the      +#
C     +              available processors.  Choose this option ONLY if the  +#
C     +              solver has been activated locally.  See subroutine     +#
C     +              (D)TD3O in file (d)subs.f for instructions on how to   +#
C     +              activate this solver and the MPI-enhancements to the   +#
C     +              conjugate gradient solver.                             +#
C     +                                                                      +#
C     + Enter ISOLVE = 1,2,3,4,5 or 6 to select a linear system solver.     +#
C     ++++++++++++++++++++++++ END OF "FINE PRINT" +++++++++++++++++++++++++#
C##############################################################################
      ISOLVE =        2
C        *******TIME-DEPENDENT PROBLEM
      itype = 2
C##############################################################################
C     Enter the initial time value (T0) and the final time value (TF), for    #
C     this time-dependent problem.  T0 defaults to 0.                         #
```

```
C                                                                         #
C     TF is not required to be greater than T0.                           #
C##########################################################################
      T0 = 0.0
C--------------------------------------------> INPUT FROM GUI <------------------
      T0 =
     & 0
C--------------------------------------------> INPUT FROM GUI <------------------
      TF =
     & 0.3
C##########################################################################
C     Is this a linear problem? ("linear" means all differential equations    #
C     and all boundary conditions are linear).  If you aren't sure, it is     #
C     safer to answer "no".                                                   #
C##########################################################################
      LINEAR = .FALSE.
C##########################################################################
C     Do you want the time step to be chosen adaptively?  If you answer       #
C     'yes', you will then be prompted to enter a value for TOLER(1), the     #
C     local relative time discretization error tolerance.  The default is     #
C     TOLER(1)=0.01.  If you answer 'no', a user-specified constant time step #
C     will be used.  We suggest that you answer 'yes' and default TOLER(1)    #
C     (although for certain linear problems, a constant time step may be much #
C     more efficient).                                                        #
C                                                                             #
C     ++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + If a negative value is specified for TOLER(1), then ABS(TOLER(1)) is +#
C     + taken to be the "absolute" error tolerance.  If a system of PDEs is  +#
C     + solved, by default the error tolerance specified in TOLER(1) applies +#
C     + to all variables, but the error tolerance for the J-th variable can  +#
C     + be set individually by specifying a value for TOLER(J) using an      +#
C     + editor, after the end of the interactive session.                    +#
C     +                                                                      +#
C     + Each time step, two steps of size dt/2 are taken, and that solution  +#
C     + is compared with the result when one step of size dt is taken.  If   +#
C     + the maximum difference between the two answers is less than the      +#
C     + tolerance (for each variable), the time step dt is accepted (and the +#
C     + next step dt is doubled, if the agreement is "too" good); otherwise  +#
C     + dt is halved and the process is repeated.  Note that forcing the     +#
C     + local (one-step) error to be less than the tolerance does not        +#
C     + guarantee that the global (cumulative) error is less than that value.+#
C     + However, as the tolerance is decreased, the global error should      +#
C     + decrease correspondingly.                                            +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++#
C##########################################################################
C--------------------------------------------> INPUT FROM GUI <------------------
      ADAPT = .FALSE.
      TOLER(1) = 0.001
      TOLER(1) =
     & 0.05
      NOUPDT = .FALSE.
C##########################################################################
C     The time stepsize will be chosen adaptively, between an upper limit     #
C     of DTMAX = (TF-T0)/NSTEPS and a lower limit of 0.0001*DTMAX.  Enter     #
C     a value for NSTEPS (the minimum number of steps).                       #
C                                                                             #
C     ++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + If you later turn off adaptive time step control, the time stepsize  +#
C     + will be constant, DT = (TF-T0)/NSTEPS.                               +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++#
C##########################################################################
C--------------------------------------------> INPUT FROM GUI <------------------
      NSTEPS =
     & 300
```

```
      dt = (tf-t0)/max(nsteps,1)
C#######################################################################
C     If you don't want to read the FINE PRINT, enter 'no'.            #
C                                                                      #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + Is the Crank-Nicolson scheme to be used to discretize time?  If you  +#
C     + answer 'no', a backward Euler scheme will be used.                   +#
C     +                                                                      +#
C     + If a user-specified constant time step is chosen, the second order   +#
C     + Crank Nicolson method is recommended only for problems with very     +#
C     + well-behaved solutions, and the first order backward Euler scheme    +#
C     + should be used for more difficult problems.  In particular, do not   +#
C     + use the Crank Nicolson method if the left hand side of any PDE is    +#
C     + zero, for example, if a mixed elliptic/parabolic problem is solved.  +#
C     +                                                                      +#
C     + If adaptive time step control is chosen, however, an extrapolation   +#
C     + is done between the 1-step and 2-step answers which makes the Euler  +#
C     + method second order, and the Crank-Nicolson method strongly stable.  +#
C     + Thus in this case, both methods have second order accuracy, and both +#
C     + are strongly stable.                                                 +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" +++++++++++++++++++++++++#
C#######################################################################
      CRANKN = .FALSE.
      FDIFF = .TRUE.
C#######################################################################
C     You may calculate one or more integrals (over the entire region) of     #
C     some functions of the solution and its derivatives.  How many integrals #
C     (NINT), if any, do you want to calculate?                               #
C                                                                             #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + In the FORTRAN program created by the preprocessor, the computed     +#
C     + values of the integrals will be returned in the vector SINT8Z.  If   +#
C     + several iterations or time steps are done, only the last computed    +#
C     + values are saved in SINT8Z (all values are printed).                 +#
C     +                                                                      +#
C     + A limiting value, SLIM8Z(I), for the I-th integral can be set        +#
C     + below in the main program.  The computations will then stop          +#
C     + gracefully whenever SINT8Z(I) > SLIM8Z(I), for any I=1...NINT.       +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" +++++++++++++++++++++++++#
C#######################################################################
C----------------------------------------> INPUT FROM GUI <-------------------
      NINT =          0
C#######################################################################
C     You may calculate one or more boundary integrals (over the entire       #
C     boundary) of some functions of the solution and its derivatives.  How   #
C     many boundary integrals (NBINT), if any, do you want to calculate?      #
C                                                                             #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + In the FORTRAN program created by the preprocessor, the computed     +#
C     + values of the integrals will be returned in the vector BINT8Z.  If   +#
C     + several iterations or time steps are done, only the last computed    +#
C     + values are saved in BINT8Z (all values are printed).                 +#
C     +                                                                      +#
C     + A limiting value, BLIM8Z(I), for the I-th boundary integral can be   +#
C     + set below in the main program.  The computations will then stop      +#
C     + gracefully whenever BINT8Z(I) > BLIM8Z(I), for any I=1...NBINT.      +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" +++++++++++++++++++++++++#
C#######################################################################
      NBINT =          0
C#######################################################################
C     If you don't want to read the FINE PRINT, enter 'no'.                   #
C                                                                             #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + Normally, interpolation is done to approximate the initial values    +#
```

```
C     + using cubic Hermites.  Since some derivatives must be interpolated,  +#
C     + if the initial values are not smooth (ie, have large or infinite     +#
C     + derivatives), the resulting cubic interpolants may have undesired    +#
C     + noise or large spikes.  Do you want to compute a least squares       +#
C     + approximation to the initial values, rather than an interpolant?     +#
C     + The least squares fit is generally much smoother, but requires one   +#
C     + extra linear system solution.                                        +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" +++++++++++++++++++++++++#
C#######################################################################
      LSQFIT = .FALSE.
C#######################################################################
C     If you don't want to read the FINE PRINT, enter 'no'.                 #
C                                                                            #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + Do you want to read the initial conditions from the restart file,    +#
C     + if it exists (and use the conditions supplied above if it does not   +#
C     + exist)?                                                              +#
C     +                                                                      +#
C     + If so, PDE2D will dump the final solution at the end of each run     +#
C     + into a restart file "pde2d.res".  Thus the usual procedure for       +#
C     + using this dump/restart option is to make sure there is no restart   +#
C     + file in your directory left over from a previous job, then the       +#
C     + first time you run this job, the initial conditions supplied above   +#
C     + will be used, but on the second and subsequent runs the restart file +#
C     + from the previous run will be used to define the initial conditions. +#
C     +                                                                      +#
C     + You can do all the "runs" in one program, by setting NPROB > 1.      +#
C     + Each pass through the DO loop, T0,TF,NSTEPS and possibly other       +#
C     + parameters may be varied, by making them functions of IPROB.         +#
C     +                                                                      +#
C     + If the 2D or 3D collocation method is used, the coordinate           +#
C     + transformation should not change between dump and restart.           +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" +++++++++++++++++++++++++#
C#######################################################################
      RESTRT = .FALSE.
C     GRIDID = .FALSE. IF FINITE ELEMENT GRID CHANGES BETWEEN DUMP, RESTART
      GRIDID = .TRUE.
C#######################################################################
C     If you do not have any periodic boundary conditions, enter IPERDC=0.   #
C                                                                            #
C     Enter IPERDC=1 for periodic conditions at P1 = P1GRID(1),P1GRID(NP1GRID)#
C           IPERDC=2 for periodic conditions at P2 = P2GRID(1),P2GRID(NP2GRID)#
C           IPERDC=4 for periodic conditions on both P1 and P2               #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + When periodic boundary conditions are selected, they apply to all    +#
C     + variables by default.  To turn off periodic boundary conditions on   +#
C     + the I-th variable, set PERDC(I) to 0 (or another appropriate value   +#
C     + of IPERDC) below in the main program and set the desired boundary    +#
C     + conditions in subroutine GB8Z, "by hand".                            +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" +++++++++++++++++++++++++#
C#######################################################################
C----------------------------------------> INPUT FROM GUI <------------------
      IPERDC =          0
C#######################################################################
C     The solution is saved on an NP1+1 by NP2+1 rectangular grid covering   #
C     the rectangle (P1A,P1B) x (P2A,P2B).  Enter values for P1A,P1B,P2A,P2B. #
C     These variables are usually defaulted.                                 #
C                                                                            #
C     The defaults are P1A = P1GRID(1), P1B = P1GRID(NP1GRID)                #
C                      P2A = P2GRID(1), P2B = P2GRID(NP2GRID)                #
C                                                                            #
C#######################################################################
C        defaults for p1a,p1b,p2a,p2b
      p1a = p1grid(1)
      p1b = p1grid(np1grid)
      p2a = p2grid(1)
      p2b = p2grid(np2grid)
C        DEFINE P1A,P1B,P2A,P2B HERE:
      call dtdpx3(np1,np2,0,p1a,p1b,p2a,p2b,zr8z,zr8z,hp18z,hp28z,hp38z,
     &p1out8z,p2out8z,p3out8z,npts8z)
C        *******allocate workspace
      call dtdpqx(np1grid,np2grid,np3grid,isolve,neqn,ii8z,ir8z,iperdc)
      if (iiwk8z.gt.1) ii8z = iiwk8z
      if (irwk8z.gt.1) ir8z = irwk8z
      allocate (iwrk8z(ii8z),rwrk8z(ir8z))
C        *******DRAW GRID LINES?
      PLOT = .TRUE.
C        *******call pde solver
      call dtdp3x(p1grid, p2grid, p3grid, np1grid,np2grid, -1, neqn, p1o
     &ut8z, p2out8z, p3out8z, uout, tout8z, npts8z, t0, dt, nsteps, nout
     &, nsave, crankn, noupdt, itype, linear, isolve, rwrk8z, ir8z, iwrk
     &8z, ii8z, iperdc, plot, lsqfit, fdiff, nint, nbint, restrt, gridid
     &)
      deallocate (iwrk8z,rwrk8z)
C        *******read from restart file to array ures8z
C       call dtdpr3(1,xres8z,nxp8z,yres8z,nyp8z,zres8z,nzp8z,ures8z,neqn)
C        *******write array ures8z back to restart file
C       call dtdpr3(2,xres8z,nxp8z,yres8z,nyp8z,zres8z,nzp8z,ures8z,neqn)
C        *******call user-written postprocessor
      call postpr(tout8z,nsave,p1out8z,p2out8z,np1,np2,uout,neqn)
C        *******CONTOUR PLOTS
C#######################################################################
C     Enter a value for IVAR, to select the variable to be plotted or  #
C     printed:                                                          #
C         IVAR = 1 means Ca  (possibly as modified by UPRINT,..)        #
C                2       Cax                                            #
C                3       Cay                                            #
C                4       CaT                                            #
C                5       CaTx                                           #
C                6       CaTy                                           #
C                7       CaC                                            #
C                8       CaCx                                           #
C                9       CaCy                                           #
C               10       CaF                                            #
C               11       CaFx                                           #
C               12       CaFy                                           #
C                .         .                                            #
C                .         .                                            #
C#######################################################################
      IVAR =          1
C#######################################################################
C     If you don't want to read the FINE PRINT, default ISET1,ISET2,ISINC.   #
C                                                                       #
C     ++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + The tabular output or plots will be made at times:            +#
C     +        T(K) = T0 + K*(TF-T0)/NSAVE                            +#
C     + for    K = ISET1, ISET1+ISINC, ISET1+2*ISINC,..., ISET2       +#
C     + Enter values for ISET1, ISET2 and ISINC.                      +#
C     +                                                               +#
C     + The default is ISET1=0, ISET2=NSAVE, ISINC=1, that is, the tabular   +#
C     + output or plots will be made at all time values for which the +#
C     + solution has been saved.                                      +#
C     ++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++#
C#######################################################################
      ISET1 = 0
      ISET2 = NSAVE
      ISINC = 1
C#######################################################################
```

```
C     If you don't want to read the FINE PRINT, enter 'no'.                   #
C                                                                             #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + Do you want to scale the axes on the plot so that the region is      +#
C     + undistorted?  Otherwise the axes will be scaled so that the figure   +#
C     + approximately fills the plot space.                                  +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++++#
C##############################################################################
      NODIST = .FALSE.
C
      ivar8z = ivar + (ivar-1)/3
      alow = amin8z(ivar8z)
      ahigh = amax8z(ivar8z)
C##############################################################################
C     Enter lower (UMIN) and upper (UMAX) bounds for the contour values. UMIN #
C     and UMAX are often defaulted.                                           #
C                                                                             #
C     Labeled contours will be drawn corresponding to the values              #
C                                                                             #
C                  UMIN + S*(UMAX-UMIN),    for S=0.05,0.15,...0.95.          #
C                                                                             #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + By default, UMIN and UMAX are set to the minimum and maximum values  +#
C     + of the variable to be plotted.  For a common scaling, you may want   +#
C     + to set UMIN=ALOW, UMAX=AHIGH.  ALOW and AHIGH are the minimum and    +#
C     + maximum values over all output points and over all saved time steps  +#
C     + or iterations.                                                       +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++++#
C##############################################################################
      UMIN = alow
      UMAX = ahigh
C##############################################################################
C     Do you want two additional unlabeled contours to be drawn between each  #
C     pair of labeled contours?                                               #
C##############################################################################
      FILLIN = .TRUE.
C##############################################################################
C     Enter a title, WITHOUT quotation marks.  A maximum of 40 characters     #
C     are allowed.  The default is no title.                                  #
C##############################################################################
      TITLE = ' '
      TITLE = 'Ca                                      '
      call dtdprx(tout8z,nsave,iset1,iset2,isinc)
      do 78756 is8z=iset1,iset2,isinc
      call dtdpln(uout8z(0,0,ivar8z,is8z),np1,np2,0,p1a,p1b,p2a,p2b,zr8z
     &,zr8z,3,ix8z,jy8z,0,title,umin,umax,nodist,fillin,tout8z(is8z),zr8
     &z,zr8z,zr8z,zr8z,2,ical8z)
78756 continue
C        *******CONTOUR PLOTS
C##############################################################################
C     Enter a value for IVAR, to select the variable to be plotted or         #
C     printed:                                                                #
C         IVAR = 1 means Ca  (possibly as modified by UPRINT,..)              #
C                2       Cax                                                  #
C                3       Cay                                                  #
C                4       CaT                                                  #
C                5       CaTx                                                 #
C                6       CaTy                                                 #
C                7       CaC                                                  #
C                8       CaCx                                                 #
C                9       CaCy                                                 #
C               10       CaF                                                  #
C               11       CaFx                                                 #
C               12       CaFy                                                 #
```

```
C                .       .                                             #
C                .       .                                             #
C#######################################################################
      IVAR =          4
C#######################################################################
C     If you don't want to read the FINE PRINT, default ISET1,ISET2,ISINC.    #
C                                                                      #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + The tabular output or plots will be made at times:              +#
C     +        T(K) = T0 + K*(TF-T0)/NSAVE                              +#
C     + for    K = ISET1, ISET1+ISINC, ISET1+2*ISINC,..., ISET2         +#
C     + Enter values for ISET1, ISET2 and ISINC.                        +#
C     +                                                                 +#
C     + The default is ISET1=0, ISET2=NSAVE, ISINC=1, that is, the tabular   +#
C     + output or plots will be made at all time values for which the        +#
C     + solution has been saved.                                        +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++#
C#######################################################################
      ISET1 = 0
      ISET2 = NSAVE
      ISINC = 1
C#######################################################################
C     If you don't want to read the FINE PRINT, enter 'no'.             #
C                                                                      #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + Do you want to scale the axes on the plot so that the region is      +#
C     + undistorted?  Otherwise the axes will be scaled so that the figure   +#
C     + approximately fills the plot space.                             +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++#
C#######################################################################
      NODIST = .FALSE.
C
      ivar8z = ivar + (ivar-1)/3
      alow = amin8z(ivar8z)
      ahigh = amax8z(ivar8z)
C#######################################################################
C     Enter lower (UMIN) and upper (UMAX) bounds for the contour values. UMIN #
C     and UMAX are often defaulted.                                    #
C                                                                      #
C     Labeled contours will be drawn corresponding to the values        #
C                                                                      #
C                 UMIN + S*(UMAX-UMIN),    for S=0.05,0.15,...0.95.     #
C                                                                      #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + By default, UMIN and UMAX are set to the minimum and maximum values  +#
C     + of the variable to be plotted.  For a common scaling, you may want   +#
C     + to set UMIN=ALOW, UMAX=AHIGH.  ALOW and AHIGH are the minimum and    +#
C     + maximum values over all output points and over all saved time steps  +#
C     + or iterations.                                                  +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++#
C#######################################################################
      UMIN = alow
      UMAX = ahigh
C#######################################################################
C     Do you want two additional unlabeled contours to be drawn between each  #
C     pair of labeled contours?                                        #
C#######################################################################
      FILLIN = .TRUE.
C#######################################################################
C     Enter a title, WITHOUT quotation marks.  A maximum of 40 characters     #
C     are allowed.  The default is no title.                           #
C#######################################################################
      TITLE = ' '
      TITLE = 'CaT                                     '
      call dtdprx(tout8z,nsave,iset1,iset2,isinc)
      do 78757 is8z=iset1,iset2,isinc
      call dtdpln(uout8z(0,0,ivar8z,is8z),np1,np2,0,p1a,p1b,p2a,p2b,zr8z
     &,zr8z,3,ix8z,jy8z,0,title,umin,umax,nodist,fillin,tout8z(is8z),zr8
     &z,zr8z,zr8z,zr8z,2,ical8z)
78757 continue
C        *******CONTOUR PLOTS
C#######################################################################
C     Enter a value for IVAR, to select the variable to be plotted or  #
C     printed:                                                          #
C         IVAR = 1 means Ca  (possibly as modified by UPRINT,..)        #
C                2       Cax                                            #
C                3       Cay                                            #
C                4       CaT                                            #
C                5       CaTx                                           #
C                6       CaTy                                           #
C                7       CaC                                            #
C                8       CaCx                                           #
C                9       CaCy                                           #
C               10       CaF                                            #
C               11       CaFx                                           #
C               12       CaFy                                           #
C                .        .                                             #
C                .        .                                             #
C#######################################################################
      IVAR =          7
C#######################################################################
C     If you don't want to read the FINE PRINT, default ISET1,ISET2,ISINC.    #
C                                                                       #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + The tabular output or plots will be made at times:             +#
C     +        T(K) = T0 + K*(TF-T0)/NSAVE                              +#
C     + for    K = ISET1, ISET1+ISINC, ISET1+2*ISINC,..., ISET2        +#
C     + Enter values for ISET1, ISET2 and ISINC.                        +#
C     +                                                                 +#
C     + The default is ISET1=0, ISET2=NSAVE, ISINC=1, that is, the tabular   +#
C     + output or plots will be made at all time values for which the  +#
C     + solution has been saved.                                        +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" +++++++++++++++++++++++++#
C#######################################################################
      ISET1 = 0
      ISET2 = NSAVE
      ISINC = 1
C#######################################################################
C     If you don't want to read the FINE PRINT, enter 'no'.            #
C                                                                       #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + Do you want to scale the axes on the plot so that the region is     +#
C     + undistorted?  Otherwise the axes will be scaled so that the figure   +#
C     + approximately fills the plot space.                            +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" +++++++++++++++++++++++++#
C#######################################################################
      NODIST = .FALSE.
C
      ivar8z = ivar + (ivar-1)/3
      alow = amin8z(ivar8z)
      ahigh = amax8z(ivar8z)
C#######################################################################
C     Enter lower (UMIN) and upper (UMAX) bounds for the contour values. UMIN #
C     and UMAX are often defaulted.                                     #
C                                                                       #
C     Labeled contours will be drawn corresponding to the values       #
C                                                                       #
C                    UMIN + S*(UMAX-UMIN),    for S=0.05,0.15,...0.95.        #
```

```
C                                                                             #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + By default, UMIN and UMAX are set to the minimum and maximum values  +#
C     + of the variable to be plotted.  For a common scaling, you may want   +#
C     + to set UMIN=ALOW, UMAX=AHIGH.  ALOW and AHIGH are the minimum and    +#
C     + maximum values over all output points and over all saved time steps  +#
C     + or iterations.                                                       +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++#
C##############################################################################
      UMIN = alow
      UMAX = ahigh
C##############################################################################
C     Do you want two additional unlabeled contours to be drawn between each  #
C     pair of labeled contours?                                               #
C##############################################################################
      FILLIN = .TRUE.
C##############################################################################
C     Enter a title, WITHOUT quotation marks.  A maximum of 40 characters     #
C     are allowed.  The default is no title.                                  #
C##############################################################################
      TITLE = ' '
      TITLE = 'CaC'
      call dtdprx(tout8z,nsave,iset1,iset2,isinc)
      do 78758 is8z=iset1,iset2,isinc
      call dtdpln(uout8z(0,0,ivar8z,is8z),np1,np2,0,p1a,p1b,p2a,p2b,zr8z
     &,zr8z,3,ix8z,jy8z,0,title,umin,umax,nodist,fillin,tout8z(is8z),zr8
     &z,zr8z,zr8z,zr8z,2,ical8z)
78758 continue
C        *******CONTOUR PLOTS
C##############################################################################
C     Enter a value for IVAR, to select the variable to be plotted or         #
C     printed:                                                                #
C         IVAR = 1 means Ca  (possibly as modified by UPRINT,..)              #
C                2       Cax                                                  #
C                3       Cay                                                  #
C                4       CaT                                                  #
C                5       CaTx                                                 #
C                6       CaTy                                                 #
C                7       CaC                                                  #
C                8       CaCx                                                 #
C                9       CaCy                                                 #
C               10       CaF                                                  #
C               11       CaFx                                                 #
C               12       CaFy                                                 #
C                .        .                                                   #
C                .        .                                                   #
C##############################################################################
      IVAR =         10
C##############################################################################
C     If you don't want to read the FINE PRINT, default ISET1,ISET2,ISINC.    #
C                                                                             #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + The tabular output or plots will be made at times:                   +#
C     +        T(K) = T0 + K*(TF-T0)/NSAVE                                   +#
C     + for    K = ISET1, ISET1+ISINC, ISET1+2*ISINC,..., ISET2              +#
C     + Enter values for ISET1, ISET2 and ISINC.                             +#
C     +                                                                      +#
C     + The default is ISET1=0, ISET2=NSAVE, ISINC=1, that is, the tabular   +#
C     + output or plots will be made at all time values for which the        +#
C     + solution has been saved.                                             +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++#
C##############################################################################
      ISET1 = 0
      ISET2 = NSAVE
      ISINC = 1
C#############################################################################
C     If you don't want to read the FINE PRINT, enter 'no'.                   #
C                                                                             #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + Do you want to scale the axes on the plot so that the region is      +#
C     + undistorted?  Otherwise the axes will be scaled so that the figure   +#
C     + approximately fills the plot space.                                  +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++++#
C#############################################################################
      NODIST = .FALSE.
C
      ivar8z = ivar + (ivar-1)/3
      alow = amin8z(ivar8z)
      ahigh = amax8z(ivar8z)
C#############################################################################
C     Enter lower (UMIN) and upper (UMAX) bounds for the contour values. UMIN #
C     and UMAX are often defaulted.                                           #
C                                                                             #
C     Labeled contours will be drawn corresponding to the values              #
C                                                                             #
C                 UMIN + S*(UMAX-UMIN),    for S=0.05,0.15,...0.95.          #
C                                                                             #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + By default, UMIN and UMAX are set to the minimum and maximum values  +#
C     + of the variable to be plotted.  For a common scaling, you may want   +#
C     + to set UMIN=ALOW, UMAX=AHIGH.  ALOW and AHIGH are the minimum and    +#
C     + maximum values over all output points and over all saved time steps +#
C     + or iterations.                                                       +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++++#
C#############################################################################
      UMIN = alow
      UMAX = ahigh
C#############################################################################
C     Do you want two additional unlabeled contours to be drawn between each  #
C     pair of labeled contours?                                               #
C#############################################################################
      FILLIN = .TRUE.
C#############################################################################
C     Enter a title, WITHOUT quotation marks.  A maximum of 40 characters     #
C     are allowed.  The default is no title.                                  #
C#############################################################################
      TITLE = ' '
      TITLE = 'CaF                                     '
      call dtdprx(tout8z,nsave,iset1,iset2,isinc)
      do 78759 is8z=iset1,iset2,isinc
      call dtdpln(uout8z(0,0,ivar8z,is8z),np1,np2,0,p1a,p1b,p2a,p2b,zr8z
     &,zr8z,3,ix8z,jy8z,0,title,umin,umax,nodist,fillin,tout8z(is8z),zr8
     &z,zr8z,zr8z,zr8z,2,ical8z)
78759 continue
C        *******SURFACE PLOTS
C#############################################################################
C     Enter a value for IVAR, to select the variable to be plotted or         #
C     printed:                                                                #
C         IVAR = 1 means Ca  (possibly as modified by UPRINT,..)              #
C                2       Cax                                                  #
C                3       Cay                                                  #
C                4       CaT                                                  #
C                5       CaTx                                                 #
C                6       CaTy                                                 #
C                7       CaC                                                  #
C                8       CaCx                                                 #
C                9       CaCy                                                 #
C               10       CaF                                                  #
```

```
C               11       CaFx                                               #
C               12       CaFy                                               #
C                .        .                                                 #
C                .        .                                                 #
C############################################################################
      IVAR =          1
C############################################################################
C     If you don't want to read the FINE PRINT, default ISET1,ISET2,ISINC.  #
C                                                                           #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + The tabular output or plots will be made at times:                 +#
C     +        T(K) = T0 + K*(TF-T0)/NSAVE                                 +#
C     + for    K = ISET1, ISET1+ISINC, ISET1+2*ISINC,..., ISET2            +#
C     + Enter values for ISET1, ISET2 and ISINC.                           +#
C     +                                                                    +#
C     + The default is ISET1=0, ISET2=NSAVE, ISINC=1, that is, the tabular +#
C     + output or plots will be made at all time values for which the      +#
C     + solution has been saved.                                           +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++#
C############################################################################
      ISET1 = 0
      ISET2 = NSAVE
      ISINC = 1
C############################################################################
C     Enter the view latitude, VLAT, and the view longitude, VLON, desired  #
C     for this plot, in degrees.  VLAT and VLON must be between 10 and 80   #
C     degrees; each defaults to 45 degrees.  VLAT and VLON are usually      #
C     defaulted.                                                            #
C############################################################################
      VLON = 45.0
      VLAT = 45.0
C
      ivar8z = ivar + (ivar-1)/3
      alow = amin8z(ivar8z)
      ahigh = amax8z(ivar8z)
C############################################################################
C     Specify the range (UMIN,UMAX) for the dependent variable axis.  UMIN  #
C     and UMAX are often defaulted.                                         #
C                                                                           #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + By default, each plot will be scaled to just fit in the plot area. +#
C     + For a common scaling, you may want to set UMIN=ALOW, UMAX=AHIGH.   +#
C     + ALOW and AHIGH are the minimum and maximum values over all output  +#
C     + points and over all saved time steps or iterations.                +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++#
C############################################################################
      UMIN = 0.0
      UMAX = 0.0
      UMIN =
     & alow
      UMAX =
     & ahigh
C############################################################################
C     Enter a title, WITHOUT quotation marks.  A maximum of 40 characters   #
C     are allowed.  The default is no title.                                #
C############################################################################
      TITLE = ' '
      TITLE = 'Ca                                      '
      call dtdprx(tout8z,nsave,iset1,iset2,isinc)
      do 78760 is8z=iset1,iset2,isinc
      call dtdplo(p1out8z,p2out8z,p3out8z,uout8z(0,0,ivar8z,is8z),np1,np
     &2,0,3,ix8z,jy8z,0,title,vlon,vlat,umin,umax,tout8z(is8z))
78760 continue
C        *******SURFACE PLOTS
```

```
C##############################################################################
C     Enter a value for IVAR, to select the variable to be plotted or         #
C     printed:                                                                 #
C         IVAR = 1 means Ca  (possibly as modified by UPRINT,..)              #
C                2       Cax                                                   #
C                3       Cay                                                   #
C                4       CaT                                                   #
C                5       CaTx                                                  #
C                6       CaTy                                                  #
C                7       CaC                                                   #
C                8       CaCx                                                  #
C                9       CaCy                                                  #
C               10       CaF                                                   #
C               11       CaFx                                                  #
C               12       CaFy                                                  #
C                .        .                                                    #
C                .        .                                                    #
C##############################################################################
      IVAR =          4
C##############################################################################
C     If you don't want to read the FINE PRINT, default ISET1,ISET2,ISINC.    #
C                                                                              #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + The tabular output or plots will be made at times:                   +#
C     +        T(K) = T0 + K*(TF-T0)/NSAVE                                   +#
C     + for    K = ISET1, ISET1+ISINC, ISET1+2*ISINC,..., ISET2              +#
C     + Enter values for ISET1, ISET2 and ISINC.                             +#
C     +                                                                      +#
C     + The default is ISET1=0, ISET2=NSAVE, ISINC=1, that is, the tabular   +#
C     + output or plots will be made at all time values for which the        +#
C     + solution has been saved.                                             +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" +++++++++++++++++++++++++#
C##############################################################################
      ISET1 = 0
      ISET2 = NSAVE
      ISINC = 1
C##############################################################################
C     Enter the view latitude, VLAT, and the view longitude, VLON, desired    #
C     for this plot, in degrees.  VLAT and VLON must be between 10 and 80     #
C     degrees; each defaults to 45 degrees.  VLAT and VLON are usually        #
C     defaulted.                                                               #
C##############################################################################
      VLON = 45.0
      VLAT = 45.0
C
      ivar8z = ivar + (ivar-1)/3
      alow = amin8z(ivar8z)
      ahigh = amax8z(ivar8z)
C##############################################################################
C     Specify the range (UMIN,UMAX) for the dependent variable axis.  UMIN    #
C     and UMAX are often defaulted.                                            #
C                                                                              #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + By default, each plot will be scaled to just fit in the plot area.   +#
C     + For a common scaling, you may want to set UMIN=ALOW, UMAX=AHIGH.     +#
C     + ALOW and AHIGH are the minimum and maximum values over all output    +#
C     + points and over all saved time steps or iterations.                  +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" +++++++++++++++++++++++++#
C##############################################################################
      UMIN = 0.0
      UMAX = 0.0
      UMIN =
     & alow
      UMAX =
     & ahigh
C#######################################################################
C     Enter a title, WITHOUT quotation marks.  A maximum of 40 characters     #
C     are allowed.  The default is no title.                                  #
C#######################################################################
      TITLE = ' '
      TITLE = 'CaT                                     '
      call dtdprx(tout8z,nsave,iset1,iset2,isinc)
      do 78761 is8z=iset1,iset2,isinc
      call dtdplo(p1out8z,p2out8z,p3out8z,uout8z(0,0,ivar8z,is8z),np1,np
     &2,0,3,ix8z,jy8z,0,title,vlon,vlat,umin,umax,tout8z(is8z))
78761 continue
C        *******SURFACE PLOTS
C#######################################################################
C     Enter a value for IVAR, to select the variable to be plotted or         #
C     printed:                                                                #
C         IVAR = 1 means Ca  (possibly as modified by UPRINT,..)              #
C                2       Cax                                                  #
C                3       Cay                                                  #
C                4       CaT                                                  #
C                5       CaTx                                                 #
C                6       CaTy                                                 #
C                7       CaC                                                  #
C                8       CaCx                                                 #
C                9       CaCy                                                 #
C               10       CaF                                                  #
C               11       CaFx                                                 #
C               12       CaFy                                                 #
C                .         .                                                  #
C                .         .                                                  #
C#######################################################################
      IVAR =          7
C#######################################################################
C     If you don't want to read the FINE PRINT, default ISET1,ISET2,ISINC.    #
C                                                                             #
C     ++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + The tabular output or plots will be made at times:                   +#
C     +        T(K) = T0 + K*(TF-T0)/NSAVE                                   +#
C     + for    K = ISET1, ISET1+ISINC, ISET1+2*ISINC,..., ISET2              +#
C     + Enter values for ISET1, ISET2 and ISINC.                             +#
C     +                                                                      +#
C     + The default is ISET1=0, ISET2=NSAVE, ISINC=1, that is, the tabular   +#
C     + output or plots will be made at all time values for which the        +#
C     + solution has been saved.                                             +#
C     ++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++#
C#######################################################################
      ISET1 = 0
      ISET2 = NSAVE
      ISINC = 1
C#######################################################################
C     Enter the view latitude, VLAT, and the view longitude, VLON, desired    #
C     for this plot, in degrees.  VLAT and VLON must be between 10 and 80     #
C     degrees; each defaults to 45 degrees.  VLAT and VLON are usually        #
C     defaulted.                                                              #
C#######################################################################
      VLON = 45.0
      VLAT = 45.0
C
      ivar8z = ivar + (ivar-1)/3
      alow = amin8z(ivar8z)
      ahigh = amax8z(ivar8z)
C#######################################################################
C     Specify the range (UMIN,UMAX) for the dependent variable axis.  UMIN    #
C     and UMAX are often defaulted.                                           #
```

```
C                                                                              #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + By default, each plot will be scaled to just fit in the plot area.   +#
C     + For a common scaling, you may want to set UMIN=ALOW, UMAX=AHIGH.     +#
C     + ALOW and AHIGH are the minimum and maximum values over all output    +#
C     + points and over all saved time steps or iterations.                  +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++#
C##############################################################################
      UMIN = 0.0
      UMAX = 0.0
      UMIN =
     & alow
      UMAX =
     & ahigh
C##############################################################################
C     Enter a title, WITHOUT quotation marks.  A maximum of 40 characters     #
C     are allowed.  The default is no title.                                  #
C##############################################################################
      TITLE = ' '
      TITLE = 'CaC                                     '
      call dtdprx(tout8z,nsave,iset1,iset2,isinc)
      do 78762 is8z=iset1,iset2,isinc
      call dtdplo(p1out8z,p2out8z,p3out8z,uout8z(0,0,ivar8z,is8z),np1,np
     &2,0,3,ix8z,jy8z,0,title,vlon,vlat,umin,umax,tout8z(is8z))
78762 continue
C        *******SURFACE PLOTS
C##############################################################################
C     Enter a value for IVAR, to select the variable to be plotted or         #
C     printed:                                                                #
C         IVAR = 1 means Ca  (possibly as modified by UPRINT,..)              #
C                2       Cax                                                  #
C                3       Cay                                                  #
C                4       CaT                                                  #
C                5       CaTx                                                 #
C                6       CaTy                                                 #
C                7       CaC                                                  #
C                8       CaCx                                                 #
C                9       CaCy                                                 #
C               10       CaF                                                  #
C               11       CaFx                                                 #
C               12       CaFy                                                 #
C                .        .                                                   #
C                .        .                                                   #
C##############################################################################
      IVAR =         10
C##############################################################################
C     If you don't want to read the FINE PRINT, default ISET1,ISET2,ISINC.    #
C                                                                             #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + The tabular output or plots will be made at times:                   +#
C     +        T(K) = T0 + K*(TF-T0)/NSAVE                                   +#
C     + for    K = ISET1, ISET1+ISINC, ISET1+2*ISINC,..., ISET2              +#
C     + Enter values for ISET1, ISET2 and ISINC.                             +#
C     +                                                                      +#
C     + The default is ISET1=0, ISET2=NSAVE, ISINC=1, that is, the tabular   +#
C     + output or plots will be made at all time values for which the        +#
C     + solution has been saved.                                             +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++#
C##############################################################################
      ISET1 = 0
      ISET2 = NSAVE
      ISINC = 1
C##############################################################################
C     Enter the view latitude, VLAT, and the view longitude, VLON, desired    #
```

```
C     for this plot, in degrees.  VLAT and VLON must be between 10 and 80     #
C     degrees; each defaults to 45 degrees.  VLAT and VLON are usually        #
C     defaulted.                                                              #
C#############################################################################
      VLON = 45.0
      VLAT = 45.0
C
      ivar8z = ivar + (ivar-1)/3
      alow = amin8z(ivar8z)
      ahigh = amax8z(ivar8z)
C#############################################################################
C     Specify the range (UMIN,UMAX) for the dependent variable axis.  UMIN    #
C     and UMAX are often defaulted.                                           #
C                                                                             #
C     ++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + By default, each plot will be scaled to just fit in the plot area.   +#
C     + For a common scaling, you may want to set UMIN=ALOW, UMAX=AHIGH.     +#
C     + ALOW and AHIGH are the minimum and maximum values over all output    +#
C     + points and over all saved time steps or iterations.                  +#
C     ++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++#
C#############################################################################
      UMIN = 0.0
      UMAX = 0.0
      UMIN =
     & alow
      UMAX =
     & ahigh
C#############################################################################
C     Enter a title, WITHOUT quotation marks.  A maximum of 40 characters     #
C     are allowed.  The default is no title.                                  #
C#############################################################################
      TITLE = ' '
      TITLE = 'CaF                                     '
      call dtdprx(tout8z,nsave,iset1,iset2,isinc)
      do 78763 is8z=iset1,iset2,isinc
      call dtdplo(p1out8z,p2out8z,p3out8z,uout8z(0,0,ivar8z,is8z),np1,np
     &2,0,3,ix8z,jy8z,0,title,vlon,vlat,umin,umax,tout8z(is8z))
78763 continue
C        *******VECTOR PLOTS
C#############################################################################
C     Enter values for IVAR1, IVAR2 to select the components Vr1 and Vr2      #
C     of the vector to be plotted.                                            #
C      IVAR1,IVAR2 = 1 means Ca  (possibly as modified by UPRINT,...)         #
C                    2       Cax                                              #
C                    3       Cay                                              #
C                    4       CaT                                              #
C                    5       CaTx                                             #
C                    6       CaTy                                             #
C                    7       CaC                                              #
C                    8       CaCx                                             #
C                    9       CaCy                                             #
C                   10       CaF                                              #
C                   11       CaFx                                             #
C                   12       CaFy                                             #
C                    .        .                                               #
C                    .        .                                               #
C     Vr1 and Vr2 are assumed to be the components of the vector in           #
C     Cartesian coordinates.                                                  #
C#############################################################################
      IVAR1 =          2
      IVAR2 =          3
C#############################################################################
C     If you don't want to read the FINE PRINT, default ISET1,ISET2,ISINC.    #
C                                                                             #
```

```
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + The tabular output or plots will be made at times:                   +#
C     +        T(K) = T0 + K*(TF-T0)/NSAVE                                   +#
C     + for    K = ISET1, ISET1+ISINC, ISET1+2*ISINC,..., ISET2              +#
C     + Enter values for ISET1, ISET2 and ISINC.                             +#
C     +                                                                      +#
C     + The default is ISET1=0, ISET2=NSAVE, ISINC=1, that is, the tabular   +#
C     + output or plots will be made at all time values for which the        +#
C     + solution has been saved.                                             +#
C     ++++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++#
C#############################################################################
      ISET1 = 0
      ISET2 = NSAVE
      ISINC = 1
C#############################################################################
C     If you don't want to read the FINE PRINT, enter 'no'.                  #
C                                                                            #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + Do you want to scale the axes on the plot so that the region is      +#
C     + undistorted?  Otherwise the axes will be scaled so that the figure   +#
C     + approximately fills the plot space.                                  +#
C     ++++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++#
C#############################################################################
      NODIST = .FALSE.
C
      ivr18z = ivar1 + (ivar1-1)/3
      ivr28z = ivar2 + (ivar2-1)/3
      a1mag = max(abs(amin8z(ivr18z)),abs(amax8z(ivr18z)))
      a2mag = max(abs(amin8z(ivr28z)),abs(amax8z(ivr28z)))
C#############################################################################
C     For the purpose of scaling the arrows, the ranges of the two components #
C     of the vector are assumed to be (-VR1MAG,VR1MAG) and (-VR2MAG,VR2MAG).  #
C     Enter values for VR1MAG and VR2MAG.  VR1MAG and VR2MAG are often        #
C     defaulted.                                                             #
C                                                                            #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + By default, VR1MAG and VR2MAG are the maxima of the absolute values  +#
C     + of the first and second components.  For a common scaling, you may   +#
C     + want to set VR1MAG=A1MAG, VR2MAG=A2MAG.  A1MAG, A2MAG are the         +#
C     + maxima of the absolute values over all output points and over all    +#
C     + saved time steps or iterations.                                      +#
C     ++++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++#
C#############################################################################
      VR1MAG = 0.0
      VR2MAG = 0.0
C#############################################################################
C     Enter a title, WITHOUT quotation marks.  A maximum of 40 characters     #
C     are allowed.  The default is no title.                                 #
C#############################################################################
      TITLE = ' '
      TITLE = 'gradient of Ca                          '
      call dtdprx(tout8z,nsave,iset1,iset2,isinc)
      do 78764 is8z=iset1,iset2,isinc
      call dtdplq(uout8z(0,0,ivr18z,is8z),uout8z(0,0,ivr28z,is8z),uout8z
     &(0,0,4,is8z),np1,np2,0,p1a,p1b,p2a,p2b,zr8z,zr8z,3,ix8z,jy8z,0,tit
     &le,vr1mag,vr2mag,zr8z,zr8z,nodist,tout8z(is8z),zr8z,zr8z,zr8z,zr8z
     &,2,ical8z)
78764 continue
C        *******VECTOR PLOTS
C#############################################################################
C     Enter values for IVAR1, IVAR2 to select the components Vr1 and Vr2      #
C     of the vector to be plotted.                                           #
C      IVAR1,IVAR2 = 1 means Ca  (possibly as modified by UPRINT,...)         #
C                    2       Cax                                             #
```

```
C                    3       Cay                                               #
C                    4       CaT                                               #
C                    5       CaTx                                              #
C                    6       CaTy                                              #
C                    7       CaC                                               #
C                    8       CaCx                                              #
C                    9       CaCy                                              #
C                   10       CaF                                               #
C                   11       CaFx                                              #
C                   12       CaFy                                              #
C                    .        .                                                #
C                    .        .                                                #
C     Vr1 and Vr2 are assumed to be the components of the vector in           #
C     Cartesian coordinates.                                                  #
C###############################################################################
      IVAR1 =          5
      IVAR2 =          6
C###############################################################################
C     If you don't want to read the FINE PRINT, default ISET1,ISET2,ISINC.    #
C                                                                             #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + The tabular output or plots will be made at times:                   +#
C     +        T(K) = T0 + K*(TF-T0)/NSAVE                                   +#
C     + for    K = ISET1, ISET1+ISINC, ISET1+2*ISINC,..., ISET2              +#
C     + Enter values for ISET1, ISET2 and ISINC.                             +#
C     +                                                                      +#
C     + The default is ISET1=0, ISET2=NSAVE, ISINC=1, that is, the tabular   +#
C     + output or plots will be made at all time values for which the        +#
C     + solution has been saved.                                             +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" +++++++++++++++++++++++++#
C###############################################################################
      ISET1 = 0
      ISET2 = NSAVE
      ISINC = 1
C###############################################################################
C     If you don't want to read the FINE PRINT, enter 'no'.                   #
C                                                                             #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + Do you want to scale the axes on the plot so that the region is      +#
C     + undistorted?  Otherwise the axes will be scaled so that the figure   +#
C     + approximately fills the plot space.                                  +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" +++++++++++++++++++++++++#
C###############################################################################
      NODIST = .FALSE.
C
      ivr18z = ivar1 + (ivar1-1)/3
      ivr28z = ivar2 + (ivar2-1)/3
      a1mag = max(abs(amin8z(ivr18z)),abs(amax8z(ivr18z)))
      a2mag = max(abs(amin8z(ivr28z)),abs(amax8z(ivr28z)))
C###############################################################################
C     For the purpose of scaling the arrows, the ranges of the two components #
C     of the vector are assumed to be (-VR1MAG,VR1MAG) and (-VR2MAG,VR2MAG).  #
C     Enter values for VR1MAG and VR2MAG.  VR1MAG and VR2MAG are often        #
C     defaulted.                                                              #
C                                                                             #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + By default, VR1MAG and VR2MAG are the maxima of the absolute values  +#
C     + of the first and second components.  For a common scaling, you may   +#
C     + want to set VR1MAG=A1MAG, VR2MAG=A2MAG.  A1MAG, A2MAG are the        +#
C     + maxima of the absolute values over all output points and over all    +#
C     + saved time steps or iterations.                                      +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" +++++++++++++++++++++++++#
C###############################################################################
      VR1MAG = 0.0
      VR2MAG = 0.0
C#######################################################################
C     Enter a title, WITHOUT quotation marks.  A maximum of 40 characters     #
C     are allowed.  The default is no title.                                   #
C#######################################################################
      TITLE = ' '
      TITLE = 'gradient of CaT                          '
      call dtdprx(tout8z,nsave,iset1,iset2,isinc)
      do 78765 is8z=iset1,iset2,isinc
      call dtdplq(uout8z(0,0,ivr18z,is8z),uout8z(0,0,ivr28z,is8z),uout8z
     &(0,0,4,is8z),np1,np2,0,p1a,p1b,p2a,p2b,zr8z,zr8z,3,ix8z,jy8z,0,tit
     &le,vr1mag,vr2mag,zr8z,zr8z,nodist,tout8z(is8z),zr8z,zr8z,zr8z,zr8z
     &,2,ical8z)
78765 continue
C        *******VECTOR PLOTS
C#######################################################################
C     Enter values for IVAR1, IVAR2 to select the components Vr1 and Vr2      #
C     of the vector to be plotted.                                             #
C      IVAR1,IVAR2 = 1 means Ca  (possibly as modified by UPRINT,...)          #
C                    2       Cax                                               #
C                    3       Cay                                               #
C                    4       CaT                                               #
C                    5       CaTx                                              #
C                    6       CaTy                                              #
C                    7       CaC                                               #
C                    8       CaCx                                              #
C                    9       CaCy                                              #
C                   10       CaF                                               #
C                   11       CaFx                                              #
C                   12       CaFy                                              #
C                    .        .                                                #
C                    .        .                                                #
C     Vr1 and Vr2 are assumed to be the components of the vector in           #
C     Cartesian coordinates.                                                   #
C#######################################################################
      IVAR1 =          8
      IVAR2 =          9
C#######################################################################
C     If you don't want to read the FINE PRINT, default ISET1,ISET2,ISINC.    #
C                                                                              #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + The tabular output or plots will be made at times:                   +#
C     +        T(K) = T0 + K*(TF-T0)/NSAVE                                   +#
C     + for    K = ISET1, ISET1+ISINC, ISET1+2*ISINC,..., ISET2              +#
C     + Enter values for ISET1, ISET2 and ISINC.                             +#
C     +                                                                      +#
C     + The default is ISET1=0, ISET2=NSAVE, ISINC=1, that is, the tabular   +#
C     + output or plots will be made at all time values for which the        +#
C     + solution has been saved.                                             +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++++#
C#######################################################################
      ISET1 = 0
      ISET2 = NSAVE
      ISINC = 1
C#######################################################################
C     If you don't want to read the FINE PRINT, enter 'no'.                   #
C                                                                              #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + Do you want to scale the axes on the plot so that the region is      +#
C     + undistorted?  Otherwise the axes will be scaled so that the figure   +#
C     + approximately fills the plot space.                                  +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++++#
C#######################################################################
      NODIST = .FALSE.
```

```
C
      ivr18z = ivar1 + (ivar1-1)/3
      ivr28z = ivar2 + (ivar2-1)/3
      a1mag = max(abs(amin8z(ivr18z)),abs(amax8z(ivr18z)))
      a2mag = max(abs(amin8z(ivr28z)),abs(amax8z(ivr28z)))
C#######################################################################
C     For the purpose of scaling the arrows, the ranges of the two components #
C     of the vector are assumed to be (-VR1MAG,VR1MAG) and (-VR2MAG,VR2MAG).  #
C     Enter values for VR1MAG and VR2MAG.  VR1MAG and VR2MAG are often        #
C     defaulted.                                                              #
C                                                                             #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + By default, VR1MAG and VR2MAG are the maxima of the absolute values  +#
C     + of the first and second components.  For a common scaling, you may   +#
C     + want to set VR1MAG=A1MAG, VR2MAG=A2MAG.  A1MAG, A2MAG are the         +#
C     + maxima of the absolute values over all output points and over all    +#
C     + saved time steps or iterations.                                      +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++#
C#######################################################################
      VR1MAG = 0.0
      VR2MAG = 0.0
C#######################################################################
C     Enter a title, WITHOUT quotation marks.  A maximum of 40 characters     #
C     are allowed.  The default is no title.                                  #
C#######################################################################
      TITLE = ' '
      TITLE = 'gradient of CaC                         '
      call dtdprx(tout8z,nsave,iset1,iset2,isinc)
      do 78766 is8z=iset1,iset2,isinc
      call dtdplq(uout8z(0,0,ivr18z,is8z),uout8z(0,0,ivr28z,is8z),uout8z
     &(0,0,4,is8z),np1,np2,0,p1a,p1b,p2a,p2b,zr8z,zr8z,3,ix8z,jy8z,0,tit
     &le,vr1mag,vr2mag,zr8z,zr8z,nodist,tout8z(is8z),zr8z,zr8z,zr8z,zr8z
     &,2,ical8z)
78766 continue
C        *******VECTOR PLOTS
C#######################################################################
C     Enter values for IVAR1, IVAR2 to select the components Vr1 and Vr2      #
C     of the vector to be plotted.                                            #
C      IVAR1,IVAR2 = 1 means Ca  (possibly as modified by UPRINT,...)         #
C                    2       Cax                                              #
C                    3       Cay                                              #
C                    4       CaT                                              #
C                    5       CaTx                                             #
C                    6       CaTy                                             #
C                    7       CaC                                              #
C                    8       CaCx                                             #
C                    9       CaCy                                             #
C                   10       CaF                                              #
C                   11       CaFx                                             #
C                   12       CaFy                                             #
C                    .        .                                               #
C                    .        .                                               #
C     Vr1 and Vr2 are assumed to be the components of the vector in           #
C     Cartesian coordinates.                                                  #
C#######################################################################
      IVAR1 =          11
      IVAR2 =          12
C#######################################################################
C     If you don't want to read the FINE PRINT, default ISET1,ISET2,ISINC.    #
C                                                                             #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + The tabular output or plots will be made at times:                   +#
C     +        T(K) = T0 + K*(TF-T0)/NSAVE                                   +#
C     + for    K = ISET1, ISET1+ISINC, ISET1+2*ISINC,..., ISET2              +#
```

```
C     + Enter values for ISET1, ISET2 and ISINC.                             +#
C     +                                                                      +#
C     + The default is ISET1=0, ISET2=NSAVE, ISINC=1, that is, the tabular   +#
C     + output or plots will be made at all time values for which the        +#
C     + solution has been saved.                                             +#
C     ++++++++++++++++++++++++++ END OF "FINE PRINT" +++++++++++++++++++++++++#
C###############################################################################
      ISET1 = 0
      ISET2 = NSAVE
      ISINC = 1
C###############################################################################
C     If you don't want to read the FINE PRINT, enter 'no'.                   #
C                                                                              #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + Do you want to scale the axes on the plot so that the region is      +#
C     + undistorted?  Otherwise the axes will be scaled so that the figure   +#
C     + approximately fills the plot space.                                  +#
C     ++++++++++++++++++++++++++ END OF "FINE PRINT" +++++++++++++++++++++++++#
C###############################################################################
      NODIST = .FALSE.
C
      ivr18z = ivar1 + (ivar1-1)/3
      ivr28z = ivar2 + (ivar2-1)/3
      a1mag = max(abs(amin8z(ivr18z)),abs(amax8z(ivr18z)))
      a2mag = max(abs(amin8z(ivr28z)),abs(amax8z(ivr28z)))
C###############################################################################
C     For the purpose of scaling the arrows, the ranges of the two components #
C     of the vector are assumed to be (-VR1MAG,VR1MAG) and (-VR2MAG,VR2MAG).  #
C     Enter values for VR1MAG and VR2MAG.  VR1MAG and VR2MAG are often        #
C     defaulted.                                                              #
C                                                                              #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + By default, VR1MAG and VR2MAG are the maxima of the absolute values  +#
C     + of the first and second components.  For a common scaling, you may   +#
C     + want to set VR1MAG=A1MAG, VR2MAG=A2MAG.  A1MAG, A2MAG are the        +#
C     + maxima of the absolute values over all output points and over all    +#
C     + saved time steps or iterations.                                      +#
C     ++++++++++++++++++++++++++ END OF "FINE PRINT" +++++++++++++++++++++++++#
C###############################################################################
      VR1MAG = 0.0
      VR2MAG = 0.0
C###############################################################################
C     Enter a title, WITHOUT quotation marks.  A maximum of 40 characters     #
C     are allowed.  The default is no title.                                  #
C###############################################################################
      TITLE = ' '
      TITLE = 'gradient of CaF                         '
      call dtdprx(tout8z,nsave,iset1,iset2,isinc)
      do 78767 is8z=iset1,iset2,isinc
      call dtdplq(uout8z(0,0,ivr18z,is8z),uout8z(0,0,ivr28z,is8z),uout8z
     &(0,0,4,is8z),np1,np2,0,p1a,p1b,p2a,p2b,zr8z,zr8z,3,ix8z,jy8z,0,tit
     &le,vr1mag,vr2mag,zr8z,zr8z,nodist,tout8z(is8z),zr8z,zr8z,zr8z,zr8z
     &,2,ical8z)
78767 continue
78755 continue
      call endgks
      stop
      end


      subroutine tran8z(itrans,p1,p2,p38z)
      implicit double precision (a-h,o-z)
      common /dtdp41/x,y,z8z,x1,x2,x3,y1,y2,y3,z1,z2,z3,x11,x21,x31,x12,
     &x22,x32,x13,x23,x33,y11,y21,y31,y12,y22,y32,y13,y23,y33,z11,z21,z3
     &1,z12,z22,z32,z13,z23,z33
      common/parm8z/pi,DCA,Umax,EC50 ,Hill,CArest,RC,Aca
     &,OD2,Slope ,Slope2,Width,TN,Flu,Cal,ktn_p,ktn_n,kcal_p
     &,kflu_n,DCal,Dflu ,DU ,OTB ,kcal_n, kflu_p
C#####################################################################
C     You can solve problems in your region only if you can describe it by    #
C                        X = X(P1,P2)                                         #
C                        Y = Y(P1,P2)                                         #
C     with constant limits on the parameters P1,P2.  If your region is        #
C     rectangular, enter ITRANS=0 and the trivial parameterization            #
C                        X = P1                                               #
C                        Y = P2                                               #
C     will be used.  Otherwise, you need to read the FINE PRINT below.        #
C                                                                             #
C     ++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + If P1,P2 represent polar or other non-Cartesian coordinates, you can +#
C     + reference the Cartesian coordinates X,Y and derivatives of your      +#
C     + unknowns with respect to these coordinates, when you define your     +#
C     + PDE coefficients, boundary conditions, and volume and boundary       +#
C     + integrals, if you enter ITRANS .NE. 0.  Enter:                       +#
C     +   ITRANS = 1, if P1,P2 are polar coordinates, that is, if           +#
C     +               P1=R, P2=Theta, where    X = R*cos(Theta)              +#
C     +                                        Y = R*sin(Theta)              +#
C     +   ITRANS = -1, same as ITRANS=1, but P1=Theta, P2=R                  +#
C     +   ITRANS = 3, to define your own coordinate transformation.  In      +#
C     +               this case, you will be prompted to define X,Y and      +#
C     +               their first and second derivatives in terms of P1,P2.  +#
C     +               Because of symmetry, you will not be prompted for all  +#
C     +               of the second derivatives.  If you make a mistake in   +#
C     +               computing any of these derivatives, PDE2D will usually +#
C     +               be able to issue a warning message. (X1 = dX/dP1, etc) +#
C     +   ITRANS = -3, same as ITRANS=3, but you will only be prompted to    +#
C     +               define X,Y; their first and second derivatives will    +#
C     +               be approximated using finite differences.              +#
C     +   When ITRANS = -3 or 3, the first derivatives of X,Y must all be    +#
C     +   continuous.                                                        +#
C     ++++++++++++++++++++++++ END OF "FINE PRINT" +++++++++++++++++++++++#
C#####################################################################
      ITRANS =          0
      return
      end


      subroutine pdes8z(yd8z,i8z,j8z,kint8z,p1,p2,p38z,t,uu8z)
      implicit double precision (a-h,o-z)

      INTEGER :: gg,jj
      REAL,DIMENSION(0:2000,0:1000)::td1,tb1,tb2,td2,tb3,td3,tb4,td4

      REAL:: mv=0

      parameter (neqnmx=  99)
C        un8z(1,I),un8z(2,I),... hold the (rarely used) values
C        of UI,UI1,... from the previous iteration or time step
      common /dtdp5x/un8z(10,neqnmx)
      common /dtdp18/norm1,norm2,n38z
      double precision norm1,norm2,n38z,normx,normy,nz8z
      dimension uu8z(10,neqnmx)
      common/parm8z/pi,DCA,Umax,EC50 ,Hill,CArest,RC,Aca
     &,OD2,Slope ,Slope2,Width,TN,Flu,Cal,ktn_p,ktn_n,kcal_p
     &,kflu_n,DCal,Dflu ,DU ,OTB ,kcal_n, kflu_p
      zr8z = 0.0
      Ca  = uu8z(1, 1)
      Ca1 = uu8z(2, 1)
      Ca2 = uu8z(3, 1)
      Ca11= uu8z(5, 1)
      Ca22= uu8z(6, 1)
      Ca12= uu8z(8, 1)
      Ca21= uu8z(8, 1)
      CaT  = uu8z(1, 2)
      CaT1 = uu8z(2, 2)
      CaT2 = uu8z(3, 2)
      CaT11= uu8z(5, 2)
      CaT22= uu8z(6, 2)
      CaT12= uu8z(8, 2)
      CaT21= uu8z(8, 2)
      CaC  = uu8z(1, 3)
      CaC1 = uu8z(2, 3)
      CaC2 = uu8z(3, 3)
      CaC11= uu8z(5, 3)
      CaC22= uu8z(6, 3)
      CaC12= uu8z(8, 3)
      CaC21= uu8z(8, 3)
      CaF  = uu8z(1, 4)
      CaF1 = uu8z(2, 4)
      CaF2 = uu8z(3, 4)
      CaF11= uu8z(5, 4)
      CaF22= uu8z(6, 4)
      CaF12= uu8z(8, 4)
      CaF21= uu8z(8, 4)
      call dtdpcd(p1,p2,p38z)
      call dtdpcb(p1,p2,p38z,norm1,norm2,n38z,x,y,z8z,normx,normy,nz8z,3
     &)
      call dtdpcc(p1,p2,p38z,
     &          Ca1,Ca2,zr8z,Ca11,Ca22,zr8z,Ca12,zr8z,zr8z,
     & x,y,z8z,Cax,Cay,uz8z,Caxx,Cayy,uzz8z,Caxy,uxz8z,uyz8z,
     & Cayx,uzx8z,uzy8z,dvol,darea)
      Canorm = Cax*normx + Cay*normy
      call dtdpcc(p1,p2,p38z,
     &          CaT1,CaT2,zr8z,CaT11,CaT22,zr8z,CaT12,zr8z,zr8z,
     & x,y,z8z,CaTx,CaTy,uz8z,CaTxx,CaTyy,uzz8z,CaTxy,uxz8z,uyz8z,
     & CaTyx,uzx8z,uzy8z,dvol,darea)
      CaTnorm = CaTx*normx + CaTy*normy
      call dtdpcc(p1,p2,p38z,
     &          CaC1,CaC2,zr8z,CaC11,CaC22,zr8z,CaC12,zr8z,zr8z,
     & x,y,z8z,CaCx,CaCy,uz8z,CaCxx,CaCyy,uzz8z,CaCxy,uxz8z,uyz8z,
     & CaCyx,uzx8z,uzy8z,dvol,darea)
      CaCnorm = CaCx*normx + CaCy*normy
      call dtdpcc(p1,p2,p38z,
     &          CaF1,CaF2,zr8z,CaF11,CaF22,zr8z,CaF12,zr8z,zr8z,
     & x,y,z8z,CaFx,CaFy,uz8z,CaFxx,CaFyy,uzz8z,CaFxy,uxz8z,uyz8z,
     & CaFyx,uzx8z,uzy8z,dvol,darea)
      CaFnorm = CaFx*normx + CaFy*normy
                          if (i8z.eq.0) then
      yd8z = 0.0
C################################################################################
C     Enter FORTRAN expressions for the functions whose integrals are to be    #
C     calculated and printed.  They may be functions of                        #
C                                                                              #
C        X,Y,Ca,Cax,Cay,Caxx,Cayy,Caxy                                         #
C            CaT,CaTx,CaTy,CaTxx,CaTyy,CaTxy                                   #
C            CaC,CaCx,CaCy,CaCxx,CaCyy,CaCxy                                   #
C            CaF,CaFx,CaFy,CaFxx,CaFyy,CaFxy and (if applicable) T             #
C             .   .   .   .   .    .    .                                      #
C                                                                              #
C     The parameters P1,P2 and derivatives with respect to these may also      #
C     be referenced (Ca1 = dCa/dP1, etc):                                      #
C           Ca1,Ca2,Ca11,Ca22,Ca12                                             #
```

```
C           CaT1,CaT2,CaT11,CaT22,CaT12                                       #
C           CaC1,CaC2,CaC11,CaC22,CaC12                                       #
C           CaF1,CaF2,CaF11,CaF22,CaF12                                       #
C             .    .    .    .    .                                           #
C                                                                             #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + If you only want to integrate a function over part of the region,    +#
C     + define that function to be zero in the rest of the region.           +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++++#
C##############################################################################
C                                                  INTEGRAL1 DEFINED
      if (kint8z.eq.1) yd8z =
     & Ca
C##############################################################################
C     Enter FORTRAN expressions for the functions whose integrals are to be   #
C     calculated and printed.  They may be functions of                       #
C                                                                             #
C        X,Y,Ca,Cax,Cay,Caxx,Cayy,Caxy                                        #
C            CaT,CaTx,CaTy,CaTxx,CaTyy,CaTxy                                  #
C            CaC,CaCx,CaCy,CaCxx,CaCyy,CaCxy                                  #
C            CaF,CaFx,CaFy,CaFxx,CaFyy,CaFxy and (if applicable) T            #
C              .    .    .    .    .    .                                     #
C                                                                             #
C     The components (NORMx,NORMy) of the unit outward normal vector          #
C     may also be referenced.                                                 #
C                                                                             #
C     The parameters P1,P2 and derivatives with respect to these may also     #
C     be referenced:                                                          #
C           Ca1,Ca2,Ca11,Ca22,Ca12                                            #
C           CaT1,CaT2,CaT11,CaT22,CaT12                                       #
C           CaC1,CaC2,CaC11,CaC22,CaC12                                       #
C           CaF1,CaF2,CaF11,CaF22,CaF12                                       #
C             .    .    .    .    .                                           #
C     You can also reference the normal derivatives Canorm,CaTnorm,CaCnorm,   #
C     CaFnorm...                                                              #
C                                                                             #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + If you only want to integrate a function over part of the boundary,  +#
C     + define that function to be zero on the rest of the boundary.         +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++++#
C##############################################################################
C                                                  BND. INTEGRAL1 DEFINED
C     if (kint8z.eq.-1) yd8z =
C    & [DEFAULT SELECTED, DEFINITION COMMENTED OUT]
      if (kint8z.gt.0) yd8z = yd8z*dvol
      if (kint8z.lt.0) yd8z = yd8z*darea
                          else
C##############################################################################
C     Now enter FORTRAN expressions to define the PDE coefficients, which     #
C     may be functions of                                                     #
C                                                                             #
C        X,Y,T,Ca,Cax,Cay,Caxx,Cayy,Caxy                                      #
C              CaT,CaTx,CaTy,CaTxx,CaTyy,CaTxy                                #
C              CaC,CaCx,CaCy,CaCxx,CaCyy,CaCxy                                #
C              CaF,CaFx,CaFy,CaFxx,CaFyy,CaFxy                                #
C                .    .    .    .    .    .                                   #
C                                                                             #
C     Recall that the PDEs have the form                                      #
C                                                                             #
C     C11*d(Ca)/dT + C12*d(CaT)/dT + C13*d(CaC)/dT + C14*d(CaF)/dT +...= F1   #
C     C21*d(Ca)/dT + C22*d(CaT)/dT + C23*d(CaC)/dT + C24*d(CaF)/dT +...= F2   #
C     C31*d(Ca)/dT + C32*d(CaT)/dT + C33*d(CaC)/dT + C34*d(CaF)/dT +...= F3   #
C     C41*d(Ca)/dT + C42*d(CaT)/dT + C43*d(CaC)/dT + C44*d(CaF)/dT +...= F4   #
C                .            .            .                 .                #
```

```
C                                                                              #
C     The parameters P1,P2 and derivatives with respect to these may also     #
C     be referenced (Ca1 = dCa/dP1, etc):                                      #
C           Ca1,Ca2,Ca11,Ca22,Ca12                                             #
C           CaT1,CaT2,CaT11,CaT22,CaT12                                        #
C           CaC1,CaC2,CaC11,CaC22,CaC12                                        #
C           CaF1,CaF2,CaF11,CaF22,CaF12                                        #
C             .    .    .    .    .    .    .    .    .                        #
C###############################################################################

C@@@ Partial Differential Equations
        o=1
        z=1
C------------------Rate constant for release generator of region #2 ----------
      Rk=3
C--Rate constant for luminal Ca2+ depletion effect on Ca2+ release of region #2
        Kc=3.2
C--Rate constant for luminal Ca2+ depletion effect on Ca2+ release of region #1
C--and #3 ----------------
      Kb=5
C-----------------variable time offset for initial trigger---------------------
        DU1=0
C-----------------Time base for initial trigger--------------------------------
        OTB1=100
C-----------------Slope1 for region #2 release generator-----------------------
        Slope1=80
C-----------------Pulse modulator for region #2 release generator--------------
        Pulse1=36
        UD=0.06
C-----------------Slope2 for region #2 release generator-----------------------
        Slope3=UD*Slope1
C-----------------Deactivating parameter for region #2 release generator-------

c***************************************************************************

      iF (x.lt.1.or.x.gt.28) THEN
C----RR1=Initial trigger , RR2= Release from region #1 and #3 , RR3= Release of
C----region #2-------
        RR1=0
        RR2=0
        RR3=0
        ELSE

C   RU= Uptake and RL=leak from the SR.
        RU=0
        RL=0
C----------- Release generator location------------------------------------
        gg=nint((x*20))
        jj=nint((y*20))

c================SERCA PUMP activation ====================================

        v2=gg-v3

         if(v2.eq.35.or.v2.eq.0.or.v2.eq.36.or.gg.eq.28)then
           v3=gg
             RU =(Umax*Ca**Hill)/(EC50**Hill + Ca**Hill)
             RL =(Umax*CArest**Hill)/(EC50**Hill + CArest**Hill)
           endif


c=== Time to reach an equlibirium before any release to set resting [Ca]===
C----------- Tequ= Time of equilibrium-----------------------------------
       Tequ=0.015
       if (t.le.Tequ) then


        RR1=0
        RR2=0
      RR3=0
       else

c============================================================================
      IF (t.lt.0.001) then
C     time initation delay . b stores time not releasing.
          b = t
         RR1 = 0.0
         nn=0

c============================================================================
      ELSE
         RR1=0
            RR3=0
C-----------tt= time counter for initial release---------------------------
         tt   = t-Tequ
C---------------- Ca release expression terms -----------------------------

         DN1 = 1.0 + exp(Slope1*(DU1-OTB1*tt))
         DN2 = 1.0 + exp(OD3 + Slope3*(DU1-OTB1*tt))

c==================Release locations of RYR=================================

        v4=gg-v5



        if(gg.eq.43.or.v4.eq.36.or.v4.eq.0.or.v4.eq.35.
     &  and.gg.lt.527)then

         v5=gg

c======Initial release at region #1=========================================


         if (y.ge.0.9.and.y.le.2) then

C---------------- Ca release expression ----------------------------------

         RR1=1200*exp(-Kc*tt)*(CArest/Ca)*(exp(-Rk*tt)/DN1-1/DN2)

         endif


          if(y.gt.0.9.and.y.lt.3) then

C--------- Threshold of activation for release generator at region #2------

             if (Ca.lt.0.107) then

                 tb1(gg,jj)=t
                 RR3=0
             else

C------------td1= Release location specific time counter at region #2-----

               td1(gg,jj)=t-tb1(gg,jj)-Tequ
               DN3 = 1.0 + exp(Slope1*(-OTB*td1(gg,jj)))
               DN4 = 1.0 + exp(OD3 + Slope3*(-OTB*td1(gg,jj)))
             if (exp(-RC*td1(gg,jj)).gt.(1/DN3 - 1/DN4)) then

                  RR3=0


              else


                   kt1 = td1(gg,jj)
C---------------- Ca release expression ----------------------------------
           RR3=1200*exp(-Kc*tt)*(CArest/Ca)*(exp(-Rk*kt1)/DN3-1/DN4)

                endif
              endif
             endif

c--------------------Region #1 release-----------------------------------
           IF(y.gt.0.9.and.y.le.2.3) then


                 if (Ca.lt.0.14) then

                    tb2(gg,jj)=t
                    RR2=0.0

                 else
C------------td2= Release location specific time counter at region #1-----

                  td2(gg,jj)=t-tb2(gg,jj)-Tequ
                      D1 = 1.0 + exp(Slope*(-OTB*td2(gg,jj)))
                      D2 = 1.0 + exp(OD2 + Slope2*(-OTB*td2(gg,jj)))


                    if (exp(-RC*td2(gg,jj)).gt.(1/D1 - 1/D2)) then

                       RR2=0


                    else
                       kt2 = td2(gg,jj)

C---------------- Ca release expression ----------------------------------
              RR2=exp(-Kb*tt)*1200*(CArest/Ca)*(exp(-RC*kt2)/D1-1/D2)

                    endif


                  endif
            ENDIF


c========================Region #3 release ================================


             IF(y.gt.3) then
            RR3=0

c ----------------------- threshold of activation for region #3------------

                 if (Ca.lt.0.14) then

                    tb4(gg,jj)=t
                    RR2=0.0
                    yp=0

                 else

C------------td4= Release location specific time counter at region #3------
                  td4(gg,jj)=t-tb4(gg,jj)-Tequ

                      D1 = 1.0 + exp(Slope*(-OTB*td4(gg,jj)))
                      D2 = 1.0 + exp(OD2 + Slope2*(-OTB*td4(gg,jj)))


                    if (exp(-RC*td4(gg,jj)).gt.(1/D1 - 1/D2)) then


                       RR2=0

                    else


                      kt4 = td4(gg,jj)
C---------------- Ca release expression -----------------------------------
               RR2=exp(-Kb*tt)*1200*(CArest/Ca)*(exp(-RC*kt4)/D1-1/D2)


                    endif


                  endif

              ENDIF
c==========================================================================



c************************************************************************
        endif

         endif

          if(RR1.lt.0.0) then
           RR1=0.0
          endif


          if(RR2.lt.0.0) then
           RR2=0.0
          endif

      endif
      ENDIF

C============================================================================
                if (j8z.eq.0) then
      yd8z = 0.0
C----------------------------------------> INPUT FROM GUI <-----------------
C                                                  C(1,1) DEFINED
      if (i8z.eq. -101) yd8z =
     & 1
C                                                  C(1,2) DEFINED
      if (i8z.eq. -102) yd8z =
     & 0
C                                                  C(1,3) DEFINED
      if (i8z.eq. -103) yd8z =
     & 0
```

```
C                                                   C(1,4) DEFINED
      if (i8z.eq. -104) yd8z =
     & 0
C-------------------------------------------> INPUT FROM GUI <------------------
C                                                   F1 DEFINED
C---------------------System of PDEs expression ------------------------------

      if (i8z.eq.    1) yd8z =
     & DCA*(Caxx+Cayy)+RL + RR1+RR2+RR3-RU-ktn_p*(TN-CaT)*Ca
     & +ktn_n*CaT -kcal_p*(Cal-CaC)*Ca+kcal_n*CaC-kflu_p*(Flu-CaF)
     & *Ca+kflu_n*CaF

C-------------------------------------------> INPUT FROM GUI <-------------------
C                                                   C(2,1) DEFINED
      if (i8z.eq. -201) yd8z =
     & 0
C                                                   C(2,2) DEFINED
      if (i8z.eq. -202) yd8z =
     & 1
C                                                   C(2,3) DEFINED
      if (i8z.eq. -203) yd8z =
     & 0
C                                                   C(2,4) DEFINED
      if (i8z.eq. -204) yd8z =
     & 0
C-------------------------------------------> INPUT FROM GUI <-------------------
C                                                   F2 DEFINED
C -----------For Troponin C---------------------------------------------------
      if (i8z.eq.    2) yd8z =
     & ktn_p*(TN-CaT)*Ca-ktn_n*CaT
C-------------------------------------------> INPUT FROM GUI <-------------------
C                                                   C(3,1) DEFINED
      if (i8z.eq. -301) yd8z =
     & 0
C                                                   C(3,2) DEFINED
      if (i8z.eq. -302) yd8z =
     & 0
C                                                   C(3,3) DEFINED
      if (i8z.eq. -303) yd8z =
     & 1
C                                                   C(3,4) DEFINED
      if (i8z.eq. -304) yd8z =
     & 0
C-------------------------------------------> INPUT FROM GUI <-------------------
C                                                   F3 DEFINED

C--------------------For Calmodulin-------------------------------------------
      if (i8z.eq.    3) yd8z =
     & DCal*(CaCxx+CaCyy)+kcal_p*(Cal-CaC)*Ca-kcal_n*CaC
C-------------------------------------------> INPUT FROM GUI <-------------------
C                                                   C(4,1) DEFINED
      if (i8z.eq. -401) yd8z =
     & 0
C                                                   C(4,2) DEFINED
      if (i8z.eq. -402) yd8z =
     & 0
C                                                   C(4,3) DEFINED
      if (i8z.eq. -403) yd8z =
     & 0
C                                                   C(4,4) DEFINED
      if (i8z.eq. -404) yd8z =
     & 1
C-------------------------------------------> INPUT FROM GUI <-------------------
C                                                   F4 DEFINED
```

```
C--------------------For Fluo4------------------------------------------------
      if (i8z.eq.    4) yd8z =
     & Dflu*(CaFxx+CaFyy)+kflu_p*(Flu-CaF)*Ca-kflu_n*CaF
                else
                endif
                          endif
      return
      end


      function u8z(i8z,p1,p2,p38z,t0)
      implicit double precision (a-h,o-z)
      common/parm8z/pi,DCA,Umax,EC50 ,Hill,CArest,RC,Aca
     &,OD2,Slope ,Slope2,Width,TN,Flu,Cal,ktn_p,ktn_n,kcal_p
     &,kflu_n,DCal,Dflu ,DU ,OTB ,kcal_n, kflu_p
      call dtdpcd(p1,p2,p38z)
      call dtdpcb(p1,p2,p38z,z18z,z28z,z38z,x,y,z8z,d18z,d28z,d38z,1)
      u8z = 0.0
C#############################################################################
C     Now the initial values must be defined using FORTRAN expressions.       #
C     They may be functions of X and Y (and the parameters P1,P2), and may    #
C     also reference the initial time T0.                                     #
C#############################################################################


C=============Initial Conditions=============================================


C-------------------------------------------> INPUT FROM GUI <------------------
C----------------------Initial Ca concentration-------------------------------
C                                                   Ca0 DEFINED
      if (i8z.eq.    1) u8z =
     & 0.1
C-------------------------------------------> INPUT FROM GUI <------------------
C----------------------Initial Ca-Troponin concentration----------------------
C                                                   CaT0 DEFINED
      if (i8z.eq.    2) u8z =
     & 6.5271
C-------------------------------------------> INPUT FROM GUI <------------------
C                                                   CaC0 DEFINED
C----------------------Initial Ca-Calmodulin concentration--------------------
      if (i8z.eq.    3) u8z =
     & 0.9677
C-------------------------------------------> INPUT FROM GUI <------------------
C                                                   CaF0 DEFINED
C----------------------Initial Ca-Fluo4 concentration-------------------------
      if (i8z.eq.    4) u8z =
     & 11.91709845

      return
      end


      subroutine gb8z(gd8z,ifac8z,i8z,j8z,p1,p2,p38z,t,uu8z)
      implicit double precision (a-h,o-z)
      parameter (neqnmx=  99)
      dimension uu8z(10,neqnmx)
C        un8z(1,I),un8z(2,I),... hold the (rarely used) values
C        of UI,UI1,... from the previous iteration or time step
      common /dtdp5x/ un8z(10,neqnmx)
      common /dtdp18/norm1,norm2,n38z
      double precision none,norm1,norm2,n38z,normx,normy,nz8z
      common/parm8z/pi,DCA,Umax,EC50 ,Hill,CArest,RC,Aca
     &,OD2,Slope ,Slope2,Width,TN,Flu,Cal,ktn_p,ktn_n,kcal_p
     &,kflu_n,DCal,Dflu ,DU ,OTB ,kcal_n, kflu_p
      none = dtdplx(2)
      zr8z = 0.0
      Ca  = uu8z(1, 1)
      Ca1 = uu8z(2, 1)
      Ca2 = uu8z(3, 1)
      CaT  = uu8z(1, 2)
      CaT1 = uu8z(2, 2)
      CaT2 = uu8z(3, 2)
      CaC  = uu8z(1, 3)
      CaC1 = uu8z(2, 3)
      CaC2 = uu8z(3, 3)
      CaF  = uu8z(1, 4)
      CaF1 = uu8z(2, 4)
      CaF2 = uu8z(3, 4)
      call dtdpcd(p1,p2,p38z)
      call dtdpcb(p1,p2,p38z,norm1,norm2,n38z,x,y,z8z,normx,normy,nz8z,3
     &)
      call dtdpcb(
     & p1,p2,p38z,Ca1,Ca2,zr8z,x,y,z8z,Cax,Cay,uz8z,2)
      Canorm = Cax*normx + Cay*normy
      call dtdpcb(
     & p1,p2,p38z,CaT1,CaT2,zr8z,x,y,z8z,CaTx,CaTy,uz8z,2)
      CaTnorm = CaTx*normx + CaTy*normy
      call dtdpcb(
     & p1,p2,p38z,CaC1,CaC2,zr8z,x,y,z8z,CaCx,CaCy,uz8z,2)
      CaCnorm = CaCx*normx + CaCy*normy
      call dtdpcb(
     & p1,p2,p38z,CaF1,CaF2,zr8z,x,y,z8z,CaFx,CaFy,uz8z,2)
      CaFnorm = CaFx*normx + CaFy*normy
      if (j8z.eq.0) gd8z = 0.0
C#######################################################################
C     Enter FORTRAN expressions to define the boundary condition functions,   #
C     which may be functions of                                               #
C                                                                             #
C              X,Y,Ca,Cax,Cay,                                                #
C                  CaT,CaTx,CaTy,                                             #
C                  CaC,CaCx,CaCy,                                             #
C                  CaF,CaFx,CaFy and (if applicable) T                        #
C                    .    .    .                                              #
C                                                                             #
C     Recall that the boundary conditions have the form                       #
C                                                                             #
C                           G1 = 0                                            #
C                           G2 = 0                                            #
C                           G3 = 0                                            #
C                           G4 = 0                                            #
C                            .   .                                            #
C     Enter NONE to indicate "no" boundary condition.                         #
C                                                                             #
C     The parameters P1,P2 and derivatives with respect to these may also     #
C     be referenced (Ca1 = dCa/dP1, etc):                                     #
C                              Ca1,Ca2                                        #
C                              CaT1,CaT2                                      #
C                              CaC1,CaC2                                      #
C                              CaF1,CaF2                                      #
C                                .    .                                       #
C     The components (NORMx,NORMy) of the unit outward normal vector          #
C     may also be referenced, as well as the normal derivatives Canorm,       #
C     CaTnorm,CaCnorm,CaFnorm...                                              #
C                                                                             #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + If "no" boundary condition is specified, the corresponding PDE is    +#
C     + enforced at points just inside the boundary (exactly on the          +#
C     + boundary, if EPS8Z is set to 0 in the main program).              +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" +++++++++++++++++++++++++#
C#######################################################################
            if (ifac8z.eq. 1) then
C#######################################################################
C                                                                       #
C     First define the boundary conditions on the face P1 = P1GRID(1).  #
C#######################################################################
                          if (j8z.eq.0) then
C-----------------------------------------> INPUT FROM GUI <------------------
C                                                  G1 DEFINED
      if (i8z.eq.    1) gd8z =
     & Ca-0.1
C-----------------------------------------> INPUT FROM GUI <------------------
C                                                  G2 DEFINED
      if (i8z.eq.    2) gd8z =
     & CaT-6.5271

C-----------------------------------------> INPUT FROM GUI <------------------
C                                                  G3 DEFINED
      if (i8z.eq.    3) gd8z =
     & CaC-0.9677
C-----------------------------------------> INPUT FROM GUI <------------------
C                                                  G4 DEFINED
      if (i8z.eq.    4) gd8z =
     & CaF-11.91709845
                          else
                          endif
            endif
            if (ifac8z.eq. 2) then
C#######################################################################
C                                                                       #
C     Now define the boundary conditions on the face P1 = P1GRID(NP1GRID).    #
C#######################################################################
                          if (j8z.eq.0) then
C-----------------------------------------> INPUT FROM GUI <------------------
C                                                  G1 DEFINED
      if (i8z.eq.    1) gd8z =
     & Ca-0.1
C-----------------------------------------> INPUT FROM GUI <------------------
C                                                  G2 DEFINED
      if (i8z.eq.    2) gd8z =
     & CaT-6.5271
C-----------------------------------------> INPUT FROM GUI <------------------
C                                                  G3 DEFINED
      if (i8z.eq.    3) gd8z =
     & CaC-0.9677
C-----------------------------------------> INPUT FROM GUI <------------------
C                                                  G4 DEFINED
      if (i8z.eq.    4) gd8z =
     & CaF-11.91709845
                          else
                          endif
            endif
            if (ifac8z.eq. 3) then
C#######################################################################
C                                                                       #
C     Now define the boundary conditions on the face P2 = P2GRID(1).    #
C#######################################################################
                          if (j8z.eq.0) then
C-----------------------------------------> INPUT FROM GUI <------------------
C                                                  G1 DEFINED
      if (i8z.eq.    1) gd8z =
     & Ca-0.1
```

```
C-----------------------------------------> INPUT FROM GUI <------------------
C                                                   G2 DEFINED
      if (i8z.eq.    2) gd8z =
     & CaT-6.5271
C-----------------------------------------> INPUT FROM GUI <------------------
C                                                   G3 DEFINED
      if (i8z.eq.    3) gd8z =
     & CaC-0.9677
C-----------------------------------------> INPUT FROM GUI <------------------
C                                                   G4 DEFINED
      if (i8z.eq.    4) gd8z =
     & CaF-11.91709845
                          else
                          endif
            endif
            if (ifac8z.eq. 4) then
C################################################################################
C                                                                              #
C     Now define the boundary conditions on the face P2 = P2GRID(NP2GRID).    #
C################################################################################
                          if (j8z.eq.0) then
C-----------------------------------------> INPUT FROM GUI <------------------
C                                                   G1 DEFINED
      if (i8z.eq.    1) gd8z =
     & Ca-0.1
C-----------------------------------------> INPUT FROM GUI <------------------
C                                                   G2 DEFINED
      if (i8z.eq.    2) gd8z =
     & CaT-6.5271
C-----------------------------------------> INPUT FROM GUI <------------------
C                                                   G3 DEFINED
      if (i8z.eq.    3) gd8z =
     & CaC-0.9677
C-----------------------------------------> INPUT FROM GUI <------------------
C                                                   G4 DEFINED
      if (i8z.eq.    4) gd8z =
     & CaF-11.91709845
                          else
                          endif
            endif
      return
      end


      subroutine pmod8z(p1,p2,p38z,t,uu8z,uprint,uxprnt,uyprnt,uzpr8z)
      implicit double precision (a-h,o-z)
      dimension uu8z(10,*),uprint(*),uxprnt(*),uyprnt(*),uzpr8z(*)
      common/dtdp14/sint(20),bint(20),slim8z(20),blim8z(20)
      common/parm8z/pi,DCA,Umax,EC50 ,Hill,CArest,RC,Aca
     &,OD2,Slope ,Slope2,Width,TN,Flu,Cal,ktn_p,ktn_n,kcal_p
     &,kflu_n,DCal,Dflu ,DU ,OTB ,kcal_n, kflu_p
      zr8z = 0.0
      Ca   = uu8z(1, 1)
      Ca1  = uu8z(2, 1)
      Ca2  = uu8z(3, 1)
      Ca11= uu8z(5, 1)
      Ca22= uu8z(6, 1)
      Ca12= uu8z(8, 1)
      Ca21= uu8z(8, 1)
      CaT  = uu8z(1, 2)
      CaT1 = uu8z(2, 2)
      CaT2 = uu8z(3, 2)
      CaT11= uu8z(5, 2)
      CaT22= uu8z(6, 2)
      CaT12= uu8z(8, 2)
      CaT21= uu8z(8, 2)
      CaC  = uu8z(1, 3)
      CaC1 = uu8z(2, 3)
      CaC2 = uu8z(3, 3)
      CaC11= uu8z(5, 3)
      CaC22= uu8z(6, 3)
      CaC12= uu8z(8, 3)
      CaC21= uu8z(8, 3)
      CaF  = uu8z(1, 4)
      CaF1 = uu8z(2, 4)
      CaF2 = uu8z(3, 4)
      CaF11= uu8z(5, 4)
      CaF22= uu8z(6, 4)
      CaF12= uu8z(8, 4)
      CaF21= uu8z(8, 4)
      call dtdpcd(p1,p2,p38z)
      call dtdpcc(p1,p2,p38z,
     &          Ca1,Ca2,zr8z,Ca11,Ca22,zr8z,Ca12,zr8z,zr8z,
     & x,y,z8z,Cax,Cay,uz8z,Caxx,Cayy,uzz8z,Caxy,uxz8z,uyz8z,
     & Cayx,uzx8z,uzy8z,dvol8z,dare8z)
      uxprnt( 1) = Cax
      uyprnt( 1) = Cay
      call dtdpcc(p1,p2,p38z,
     &          CaT1,CaT2,zr8z,CaT11,CaT22,zr8z,CaT12,zr8z,zr8z,
     & x,y,z8z,CaTx,CaTy,uz8z,CaTxx,CaTyy,uzz8z,CaTxy,uxz8z,uyz8z,
     & CaTyx,uzx8z,uzy8z,dvol8z,dare8z)
      uxprnt( 2) = CaTx
      uyprnt( 2) = CaTy
      call dtdpcc(p1,p2,p38z,
     &          CaC1,CaC2,zr8z,CaC11,CaC22,zr8z,CaC12,zr8z,zr8z,
     & x,y,z8z,CaCx,CaCy,uz8z,CaCxx,CaCyy,uzz8z,CaCxy,uxz8z,uyz8z,
     & CaCyx,uzx8z,uzy8z,dvol8z,dare8z)
      uxprnt( 3) = CaCx
      uyprnt( 3) = CaCy
      call dtdpcc(p1,p2,p38z,
     &          CaF1,CaF2,zr8z,CaF11,CaF22,zr8z,CaF12,zr8z,zr8z,
     & x,y,z8z,CaFx,CaFy,uz8z,CaFxx,CaFyy,uzz8z,CaFxy,uxz8z,uyz8z,
     & CaFyx,uzx8z,uzy8z,dvol8z,dare8z)
      uxprnt( 4) = CaFx
      uyprnt( 4) = CaFy
C##############################################################################
C     If you don't want to read the FINE PRINT, default all of the following  #
C     variables.                                                               #
C                                                                              #
C     +++++++++++++++ THE "FINE PRINT" (CAN USUALLY BE IGNORED) ++++++++++++++#
C     + Normally, PDE2D saves the values of Ca,Cax,Cay,CaT,CaTx,CaTy,         +#
C     + CaC,CaCx,CaCy,CaF,CaFx,CaFy...at the output points.  If different    +#
C     + variables are to be saved (for later printing or plotting) the        +#
C     + following functions can be used to re-define the output variables:   +#
C     +    define UPRINT(1) to replace  Ca                                     +#
C     +           UXPRNT(1)             Cax                                    +#
C     +           UYPRNT(1)             Cay                                    +#
C     +           UPRINT(2)             CaT                                    +#
C     +           UXPRNT(2)             CaTx                                   +#
C     +           UYPRNT(2)             CaTy                                   +#
C     +           UPRINT(3)             CaC                                    +#
C     +           UXPRNT(3)             CaCx                                   +#
C     +           UYPRNT(3)             CaCy                                   +#
C     +           UPRINT(4)             CaF                                    +#
C     +           UXPRNT(4)             CaFx                                   +#
C     +           UYPRNT(4)             CaFy                                   +#
C     +                  .               .                                     +#
C     +                  .               .                                     +#
```

```
C     + Each function may be a function of                                   +#
C     +                                                                      +#
C     +    X,Y,Ca,Cax,Cay,Caxx,Cayy,Caxy                                     +#
C     +        CaT,CaTx,CaTy,CaTxx,CaTyy,CaTxy                               +#
C     +        CaC,CaCx,CaCy,CaCxx,CaCyy,CaCxy                               +#
C     +        CaF,CaFx,CaFy,CaFxx,CaFyy,CaFxy and (if applicable) T         +#
C     +            .   .    .    .    .     .                                +#
C     +                                                                      +#
C     + Each may also be a function of the integral estimates SINT(1),...,   +#
C     + BINT(1),...                                                          +#
C     +                                                                      +#
C     + The parameters P1,P2 and derivatives with respect to these may also  +#
C     + be referenced (Ca1 = dCa/dP1, etc):                                  +#
C     +       Ca1,Ca2,Ca11,Ca22,Ca12                                         +#
C     +       CaT1,CaT2,CaT11,CaT22,CaT12                                    +#
C     +       CaC1,CaC2,CaC11,CaC22,CaC12                                    +#
C     +       CaF1,CaF2,CaF11,CaF22,CaF12                                    +#
C     +         .    .    .    .     .                                       +#
C     +                                                                      +#
C     + The default for each variable is no change, for example, UPRINT(1)   +#
C     + defaults to Ca.  Enter FORTRAN expressions for each of the           +#
C     + following functions (or default).                                    +#
C     +++++++++++++++++++++++++ END OF "FINE PRINT" ++++++++++++++++++++++++#
C#############################################################################
C        DEFINE UPRINT(*),UXPRNT(*),UYPRNT(*) HERE:
      return
      end


      function axis8z(i8z,p1,p2,p38z,ical8z)
      implicit double precision (a-h,o-z)
      call dtdpcd(p1,p2,p38z)
      call dtdpcb(p1,p2,p38z,z18z,z28z,z38z,x,y,z8z,d18z,d28z,d38z,1)
      if (i8z.eq.1) axis8z = x
      if (i8z.eq.2) axis8z = y
      return
      end
C        dummy routines
      subroutine xy8z(i8z,iarc8z,s,x,y,s0,sf)
      implicit double precision (a-h,o-z)
      return
      end
      subroutine dis8z(x,y,ktri,triden,shape)
      implicit double precision (a-h,o-z)
      return
      end
      function fb8z(i8z,iarc8z,ktri,s,x,y,t)
      implicit double precision (a-h,o-z)
      fb8z = 0
      return
      end


      subroutine postpr(tout,nsave,p1out,p2out,np1,np2,uout,neqn)
      implicit double precision (a-h,o-z)
      dimension p1out(0:np1,0:np2),p2out(0:np1,0:np2),tout(0:nsave)
      dimension uout(0:np1,0:np2,4,neqn,0:nsave)
      common/parm8z/pi,DCA,Umax,EC50 ,Hill,CArest,RC,Aca
     &,OD2,Slope ,Slope2,Width,TN,Flu,Cal,ktn_p,ktn_n,kcal_p
     &,kflu_n,DCal,Dflu ,DU ,OTB ,kcal_n, kflu_p
      common /dtdp27/ itask,npes,icomm
      common /dtdp46/ eps8z,cgtl8z,npmx8z,itype,near8z
      data lun,lud/0,47/
      if (itask.gt.0) return
```

```
C     UOUT(I,J,IDER,IEQ,L) = U-sub-IEQ,  if IDER=1
C                            Ux-sub-IEQ, if IDER=2
C                            Uy-sub-IEQ, if IDER=3
C       (possibly as modified by UPRINT,..)
C       at the point (P1OUT(I,J) , P2OUT(I,J))
C       at time/iteration TOUT(L).
C       ******* ADD POSTPROCESSING CODE HERE:
C       IN THE EXAMPLE BELOW, MATLAB PLOTFILES pde2d.m,
C      pde2d.rdm CREATED (REMOVE  COMMENTS TO ACTIVATE)
      if (lun.eq.0) then
         lun = 46
         open (lun,file='pde2d.m')
         open (lud,file='pde2d.rdm')
         write (lun,*) 'fid = fopen(''pde2d.rdm'');'
      endif
      do 78753 l=0,nsave
         if (tout(l).ne.dtdplx(2)) nsave0 = l
78753 continue
      write (lud,78754) nsave0
      write (lud,78754) neqn
      write (lud,78754) np1
      write (lud,78754) np2
78754 format (i8)
      do 78756 i=0,np1
      do 78755 j=0,np2
         p1 = p1out(i,j)
         p2 = p2out(i,j)
         p38z = 0.0
         call dtdpcd(p1,p2,p38z)
         call dtdpcb(p1,p2,p38z,z18z,z28z,z38z,x,y,z8z,
     &   d18z,d28z,d38z,1)
         write (lud,78762) p1,p2,x,y
78755 continue
78756 continue
      do 78761 l=0,nsave0
         write (lud,78762) tout(l)
         do 78760 ieq=1,neqn
         do 78759 ider=1,3
         do 78758 i=0,np1
         do 78757 j=0,np2
            write (lud,78762) uout(i,j,ider,ieq,l)
78757    continue
78758    continue
78759    continue
78760    continue
78761 continue
78762 format (e16.8)
      write (lun,*) '% Read solution from pde2d.rdm'
      write (lun,*) 'NSAVE = fscanf(fid,''%g'',1);'
      write (lun,*) 'NEQN = fscanf(fid,''%g'',1);'
      write (lun,*) 'NP1 = fscanf(fid,''%g'',1);'
      write (lun,*) 'NP2 = fscanf(fid,''%g'',1);'
      if (itype.eq.2) then
         write (lun,*) 'L0 = 0;'
      else
         write (lun,*) 'L0 = 1;'
      endif
      write (lun,*) 'T = zeros(NSAVE+1,1);'
      write (lun,*) 'P1 = zeros(NP2+1,NP1+1);'
      write (lun,*) 'P2 = zeros(NP2+1,NP1+1);'
      write (lun,*) 'X = zeros(NP2+1,NP1+1);'
      write (lun,*) 'Y = zeros(NP2+1,NP1+1);'
      write (lun,*) 'U = zeros(NP2+1,NP1+1,NSAVE+1,3,NEQN);'
      write (lun,*) 'for i=0:NP1'
```

```
 write (lun,*) 'for j=0:NP2'
 write (lun,*) '   P1(j+1,i+1) = fscanf(fid,''%g'',1);'
 write (lun,*) '   P2(j+1,i+1) = fscanf(fid,''%g'',1);'
 write (lun,*) '   X(j+1,i+1) = fscanf(fid,''%g'',1);'
 write (lun,*) '   Y(j+1,i+1) = fscanf(fid,''%g'',1);'
 write (lun,*) 'end'
 write (lun,*) 'end'
 write (lun,*) 'for l=0:NSAVE'
 write (lun,*) 'T(l+1) = fscanf(fid,''%g'',1);'
 write (lun,*) 'for ieq=1:NEQN'
 write (lun,*) 'for ider=1:3'
 write (lun,*) 'for i=0:NP1'
 write (lun,*) 'for j=0:NP2'
 write (lun,*)
& '   U(j+1,i+1,l+1,ider,ieq) = fscanf(fid,''%g'',1);'
 write (lun,*) 'end'
 write (lun,*) 'end'
 write (lun,*) 'end'
 write (lun,*) 'end'
 write (lun,*) 'end'
 write (lun,*) 'xmin = min(min(X(:,:)));'
 write (lun,*) 'xmax = max(max(X(:,:)));'
 write (lun,*) 'ymin = min(min(Y(:,:)));'
 write (lun,*) 'ymax = max(max(Y(:,:)));'
 write (lun,*) 'hx = 0.1*(xmax-xmin);'
 write (lun,*) 'hy = 0.1*(ymax-ymin);'
 write (lun,*) '% Surface plots of each variable'
 write (lun,*) 'for IEQ=1:NEQN'
 write (lun,*) 'IDER = 1;'
 write (lun,*)
& 'umin = min(min(min(U(:,:,L0+1:NSAVE+1,IDER,IEQ))));'
 write (lun,*)
& 'umax = max(max(max(U(:,:,L0+1:NSAVE+1,IDER,IEQ))));'
 write (lun,*) 'for L=L0:NSAVE'

 write (lun,*) '   surf(X,Y,U(:,:,L+1,IDER,IEQ))'
 write (lun,*) '   colorbar'
 write (lun,*) '   axis([xmin xmax ymin ymax umin umax])'
 write (lun,*) '   xlabel(''X'')'
 write (lun,*) '   ylabel(''Y'')'
 write (lun,*) '   zlabel([''U'',num2str(IEQ)])'
 write (lun,*) '   title(['' T = '',num2str(T(L+1))])'
 write (lun,*) '   view(50,40.0)'
 write (lun,*) '   caxis([0.1 0.3])'
 write (lun,*) '   saveas(gcf,[''wave'',num2str(L),''.jpg'']); '
 write (lun,*) 'end'
 write (lun,*) 'end'

 return
 end
```